Isogeometric finite element methods for shape optimization

Vom Fachbereich Mathematik der Universität Kaiserslautern zur Verleihung des akademischen Grades

Doktor der Naturwissenschaften (Doctor rerum naturalium, Dr. rer. nat.)

genehmigte Dissertation

von

Daniela Kornelia Fußeder

1. Gutachter:	Prof. Dr. Bernd Simeon (Technische Universität Kaiserslautern)
2. Gutachter:	Prof. Dr. Jens Gravesen (Technical University of Denmark)
Disputation:	29. Oktober 2015

D 386

TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

TERRIFIC
Enhancing Interoperability

$(CM)^2$

Financial support by the European Union within the 7th Framework Programme, project FP7-2011-NMP-ICT-FoF 284981 “TERRIFIC”, and by the Center for Mathematical and Computational Modelling $(CM)^2$ in Kaiserslautern is gratefully acknowledged.

Zusammenfassung

In dieser Arbeit wird Formoptimierung mit Isogeometrischer Analysis (IGA) kombiniert und insbesondere ein abstrakter Rahmen im *optimize first–discretize then* Ansatz entwickelt. Für die Diskretisierung der Zustandsgleichung verwenden wir IGA und für den Kontrollraum bzw. für die zulässigen Formen benützen wir ebenso B-splines oder NURBS. Dies bietet uns eine große Klasse von Funktionen, um optimale Designs zu repräsentieren. Für Gradienten-basierte Optimierungsmethoden brauchen wir sogenannte Formgradienten, die sowohl als Abbruchskriterien als auch Suchrichtungen dienen und isogeometrisch bestimmt werden. Die numerische Behandlung erfordert dafür Löser für die partiellen Differentialgleichungen der Zustandsgleichung und Algorithmen zur Optimierung, wodurch Diskretisierungsfehler entstehen. Daher liegt unser Hauptaugenmerk auf dem abstrakten Rahmen für isogeometrische Formoptimierung für die spätere Implementierung und Fehleranalyse. Die enge Verbindung zwischen IGA und Geometriedarstellungen erlaubt es uns, Geometrie- und Simulation gleichermaßen mit B-splines zu diskretisieren und zu verfeinern. Numerische Beispiele belegen dann, dass dieser Ansatz auch praktisch funktioniert und Fallstudien zeigen die Verwendung von lokaler Verfeinerung.

Abstract

In this thesis we develop a shape optimization framework for isogeometric analysis in the *optimize first–discretize then* setting. For the discretization we use isogeometric analysis (IGA) to solve the state equation, and search optimal designs in a space of admissible B-spline or NURBS combinations. Thus a quite general class of functions for representing optimal shapes is available. For the gradient-descent method, the shape derivatives indicate both stopping criteria and search directions and are determined isogeometrically. The numerical treatment requires solvers for partial differential equations and optimization methods, which introduces numerical errors. The tight connection between IGA and geometry representation offers new ways of refining the geometry and analysis discretization by the same means. Therefore, our main concern is to develop the *optimize first* framework for isogeometric shape optimization as ground work for both implementation and an error analysis. Numerical examples show that this ansatz is practical and case studies indicate that it allows local refinement.

Nothing is lost, everything is transformed.
– Antoine Lavoisier

Danksagung

An dieser Stelle möchte ich allen danken, die mir während der Promotionszeit mit Rat und Tat beigestanden haben.
Zu vorderst gehört mein Dank Prof. Dr. Bernd Simeon, der mir diese Arbeit in seiner Forschungsgruppe ermöglicht hat. Erst durch seine Ermutigung, aber auch Herausforderung, zu experimentieren und Neues zu wagen, bin ich schließlich hier angekommen.
Mein Dank geht auch an Prof. Dr. Jens Gravesen für seine Bereitschaft zur Begutachtung meiner Arbeit. Die kritische, motivierende und beratende, inspirierende Begleitung als Mentor auf diesem Weg durch Dr. Anh-Vu Vuong schätze ich besonders. Einige Puzzlestücke dieser Arbeit haben sich erst durch die Diskussionen zur Formoptimierung mit Prof. Dr. Nicolas Gauger, Prof. Dr. René Pinnau und Dr. Bernhard Kiniger zusammengefügt; Dr. Utz Wever von Siemens hat mir dazu industrielle "Verbindungsteilchen" gezeigt. Für diese Blickwinkel und ihre Hilfe danke ich ihnen ganz herzlich. Bedanken möchte ich mich auch bei Dr. Christian Stinner, der immer ein offenes Ohr für Diskussionen über Sobolev-Räume hatte, bei Dr. Ronny Bergmann für seine Geduld bei Gedankenexperimenten und seine LaTeX-Tipps, bei Philip Trautmann für die Skype-Sessions zu Optimal Control und bei Dr. Julia Vuong, für ihre Unterstützung vor allem in der Abschlussphase. Für die Durchsicht der Arbeit bin ich ihr, Anh-Vu Vuong und meinen Doktorandenkollegen Dr. Oliver Weeger und Dr. Anmol Goyal sehr dankbar.
Ich danke auch meinen Kollegen Mané Harutyunyan, Dennis Merkert und Dr. Stefanie Sonner für die gute Arbeitsatmosphäre.
Meinen Eltern danke ich für ihre dauernde Unterstützung und meiner Schwester, Veronika Fußeder:

> *My head just feels in pain*
> *I missed the bus and there'll be hell today,*
> *I'm late for work again and even if i'm there,*
> *They'll all imply that i might not last the day*
> *And then you call me and it's not so bad*

[Dido, 2001]

Contents

List of Symbols

Acronyms and abbreviations

CAGD computer aided geometric design

FEM finite element method

IGA isogeometric analysis

KKT Karush-Kuhn-Tucker (optimality system)

MMA method of moving asymptotes

PDE partial differential equation

SQP sequential quadratic programming

SOP shape optimization problem

Function spaces

$\mathcal{C}^k(\Omega; Y)$ continuously differentiable functions from Ω to Y, after Definition 2.2

$\mathcal{C}^m$ continuously differentiable functions, Definition 2.1

$\langle \cdot, \cdot \rangle_{X^*,X}$ dual pairing, Section B.2.2(vii)

X^* dual space of X, Section B.2.2(vi)

$\mathcal{C}^{m,\lambda}$ Hölder space, Definition 2.3

$\mathcal{C}_0^\infty$ infinitely differentiable functions with compact support, Definition 2.1

$\mathcal{C}^\infty$ infinitely differentiable functions, Definition 2.2

L^p Lebesgue space, see equations (2.2) and (2.4)

$\mathcal{P}_{T,p}$ polynomials, multivariate, see equation (4.1)

$\mathcal{P}_{Q,p}$ polynomials, multivariate, see equation (4.2)

$\mathcal{P}_p$ polynomials, Section 4.1.1

H^k, H_0^k, $(H^k)^m$ Sobolev spaces $W^{k,2}$, see equation (2.9)

$W^{k,p}$, $W_0^{k,p}$ Sobolev spaces, see equations (2.6) and (2.11)

Splines

$\mathcal{B}(\Xi, p)$ basis for B-splines, see equation (4.5)

$\mathcal{B}(W, \Xi, p)$ basis for NURBS, Definition 4.5

X control point of NURBS/B-spline geometry, page 60

C_X	control polygon, see equation (4.29)
n	dimension of a spline space, degrees of freedom, Definition 4.2
$\hat{d}$	dimension of parameter domain $\hat{\Omega} \subset \mathbb{R}^{\hat{d}}$, page 60
G	geometry function in IGA, see equation (4.17)
γ_i	Greville point, Definition 4.22
ξ_i	knot in knot vector, Definition 4.2
Ξ	knot vector, Definition 4.2
μ	multiplicity of a knot in knot vector Ξ, Definition 4.9
$\hat{\Omega}$	parameter domain, unit hypercube in IGA, page 60
H	perspective map for homogeneous coordinates, Definition 4.20
υ_i	unique knot in Υ, Definition 4.9
Υ	unique knots of knot vector Ξ, Definition 4.9
$\mathbf{X}$	vector of all control points, Remark 4.19
$\mathbf{N}$	vector of all B-spline basis functions, Remark 4.19
N_i	B-spline basis function, non-rational: uni- or multivariate, Remark 4.8
$N_{i,p}$	B-spline basis function, non-rational: univariate, Definition 4.2
R_i	B-spline basis function, rational: NURBS, uni- or multivariate, Remark 4.8
$R_{i,p}$	B-spline basis function, rational: NURBS, univariate, Definition 4.5
$\mathcal{S}(W, \Xi, p)$	B-spline space, rational: univariate NURBS space, Definition 4.5
$\mathcal{S}(W_1, \Xi_1, p_1; \ldots; W_d, \Xi_d, p_d)$	B-spline space, rational: Tensor product like NURBS space, Definition 4.6
$\mathcal{S}$	B-spline space: rational or non-rational and univariate or multivariate
$\mathcal{S}(\Xi_1, p_1; \ldots; \Xi_d, p_d)$	B-spline space: Tensor product, Definition 4.4
$\mathcal{S}(\Xi, p)$	B-spline space: univariate, see equation (4.5)
$\mathcal{N}$	NURBS space with free weights, Definition 4.21
w	NURBS weight function, see equation (4.7)
W	NURBS weight vector, Definition 4.5
ω_i	NURBS weight, Definition 4.5

Optimization

z	adjoint state, see equation (2.57)
$\mathcal{W}_{\text{ad}}$	admissible set, see equation (2.45)
$\mathcal{Q}_{\text{ad}}$	control space, admissible, Section 2.4.1
$\mathcal{Q}$	control space, Section 2.4.1
q	control, Section 2.4.1
j	cost functional, reduced, Definition 2.26
$\hat{J}$	cost functional, transformed on parameter domain $\hat{\Omega}$, see equation (3.39)
J	cost or objective functional, see equation (1.1)
D	holding-all, Definition 3.2
$\mathcal{L}$	Lagrange functional, Definition 2.27
$\mathscr{D}^k_{\Omega_0}$	perturbation of identity: set of domains, shape space, see equation (3.6)
$\mathscr{C}^k$	perturbation of identity: space of perturbations of identity, see equation (3.5)
$\mathscr{V}^k$	perturbation of identity: space of transformations, see equation (3.4)
$\mathcal{O}_{\text{ad}}$	set of shapes, admissible, page 30
$\mathcal{O}$	set of shapes, shape "space", page 30
S	solution operator, see equation (2.21)

Simulation

$\mathcal{U}$	Banach space for PDE operator, domain, page 31
$\mathcal{Z}$	Banach space for PDE operator, range, page 31
$\hat{\mathcal{U}}$	Banach space of transformed PDE operator, domain, see equation (3.39)
$\hat{\mathcal{Z}}$	Banach space of transformed PDE operator, range, see equation (3.39)
Γ, Γ_N, Γ_D	boundaries of Ω, with boundary conditions, see equations (2.17), (2.16)
g_D	boundary conditions, Dirichlet, see equation (2.16)
g_N	boundary conditions, Neumann, see equation (2.17)
d	dimension of geometry $\Omega \subset \mathbb{R}^d$, usually $d = 1, 2, 3$
m	dimension of PDE vector-valued solution, see equation (2.14)
Λ_i	dual basis of B-splines, see equation (4.54)
$\hat{\Pi}_h$	elements in the support of a basis function in IGA, see equation (4.50)

$\mathcal{V}_h$, $\hat{\mathcal{V}}_h$ Galerkin projection space, Sections 4.3.1 and 4.3.2

$\hat{\imath}_h$ interpolation operator in IGA, see equation (4.53)

Q isogeometric element in parameter domain, see equation (4.47)

$\hat{\mathcal{K}}_h$ isogeometric mesh in parameter domain, see equation (4.48)

μ Lamé parameter (second), shear modulus, see equation (2.22)

Δ Laplace operator, Problem 1, page 11

C linear elasticity: elasticity matrix, Problems 2, 3, page 13

λ linear elasticity: Lamé parameter (first), see equation (2.22)

ν linear elasticity: Poisson ratio, Problem 2, page 12

ε, $\underline{\varepsilon}$ linear elasticity: strain, see equation (2.20)

σ, $\underline{\sigma}$ linear elasticity: stress, see equation (2.21)

L linear elliptic PDE operator of second order, see equation (2.12)

$\mathcal{V}$, $\hat{\mathcal{V}}$ test function space, Sections 4.3.1 and 4.3.2

E Young's modulus, Problem 2, page 12

e PDE operator, see equation (1.1) or page 13

ê PDE operator, transformed on parameter domain $\hat{\Omega}$, see equation (3.39)

Other mathematical symbols

ess sup essential supremum, see equation (2.4)

id identity map

I identity matrix

d metric

α multi-index, pages 8 and 60

$B_X(x, r)$ open ball of radius r around x in metric space (X, d), page 7

n outward pointing unit normal

$\mathscr{P}(X)$ power set of X

e_k standard unit vector

supp support of a function, Definition 2.2

tr trace operator

1 Introduction

An aerospace company advertises on its web page the efficiency of their airplane which consumes 20 percent less fuel than similar sized planes, [Boeing, 2015]. Talking about a fuel consumption of roughly 850 liters per 100 km on transatlantic flights, a huge saving is thus achieved by advanced aerodynamics, better turbine designs and a lightweight structure. In short, it is the very prototype of industrial shape optimization, whose aim is a shape or domain, like the profile of an airplane wing, which optimizes a given objective, like the uplift of the airfoil. Rich in real life examples, shape optimization has high industrial relevance because its ultimate goal is better performance: less material, more stability, lower failure rates, or higher output. Applications comprise designing wings of airplanes with more lift [Schmidt et al., 2011], stronger ship hulls to resist waves [Ginnis et al., 2013], and stabler bridges [Bendsøe and Sigmund, 2003]. The cost or objective under consideration depends on a shape and also on the solution of a partial differential equation (PDE). The solution to that PDE is called state and depends itself on the shape: Changing the wing profile results in a change of uplift. Thus, the shape is a control which is coupled to the state in the PDE, what makes shape optimization a special kind of optimal control problem.

Since for most partial differential equations we do not have a closed-form solution, we solve them numerically. Likewise, one may show that a solution to an optimization problem exists, but often has to employ computer algorithms to find it. Hence, to solve shape optimization problems, both the PDE and shape space are discretized, and an optimization method is used to find an optimal shape iteratively. One method of solving PDEs numerically is isogeometric analysis (IGA) which has tight links to computer aided geometric design (CAGD) and therefore seems destined for shape optimization: IGA solves the PDE on the given CAGD model where in contrast finite elements use a polygonal approximation. Thus there is a conversion between the two geometry modells back and forth, in the worst case introducing a consistency error. Since the optimization process is very delicate, an advantage of IGA in shape optimization is that it eliminates this discrepancy. In this work we develop an isogeometric framework for shape optimization problems (SOPs) with partial differential equations

$$\min J(\Omega, u) \quad \text{s.t.} \;\; \mathrm{e}(\Omega, u) = 0\,, \quad \Omega \in \mathcal{O}_{\mathrm{ad}} \tag{1.1}$$

where J is a real valued cost functional depending on a domain Ω from a set of admissible shapes $\mathcal{O}_{\mathrm{ad}}$ and the solution u of a linear elliptic PDE on Ω, which is given by the term $\mathrm{e}(\Omega, u) = 0$.

Isogeometric shape optimization is a relatively new combination; the first publication appeared in 2008. Therefore, it seems worthwhile to evaluate existing SOP approaches in the light of IGA to be able to compare their theoretical and practical performance with classical FEM shape optimization. Theoretical studies, such as a convergence analysis, use the infinite-dimensional problem, and concepts from optimal control to analyze the SOP. For IGA we develop a similar formulation in this thesis: Isogeometric analysis

transforms the PDE to a parameter domain before finding approximate solutions. This key idea is utilized in our abstract framework to state also the SOPs over this fixed parameter domain, and search for optimal transformations instead. In particular, we derive optimality conditions by considering the infinite-dimensional, transformed problem. Then, our formulation makes an isogeometric discretization of state and control directly applicable and exposes its influence on the error analysis. We compare this approach to a second one, which first discretizes (1.1) with isogeometric analysis and only then derives the optimality conditions. Eventually, we turn both approaches into a mutual, comprehensive isogeometric shape optimization algorithm where we also address practical questions like domain updates.

In the following, we comment on the different tasks adressed and retrieve the structure for this thesis.

Isogeometric analysis IGA is a Galerkin method to approximate a PDE $\mathrm{e}(\Omega, u) = 0$. It was introduced in the seminal paper [Hughes et al., 2005] to avoid the tedious task of fitting forms from computer aided design to traditional finite element meshes by working with the same geometry model throughout. In particular, it combines the fundamental idea of the finite element method (FEM) with spline techniques from CAGD for a common description of the domain and the projection space: The CAGD representation of the design is given by a B-spline or NURBS parameterization. This is used to transform the PDE to a parameter domain. In addition to such an isoparametric scheme, the simulation space is discretized by the (same) B-spline or NURBS basis functions. In [Beirão da Veiga et al., 2014] an analysis of IGA with error estimates is provided, and in [Vuong et al., 2011] a means for local adaptive refinement is presented.

Shape optimization The challenging trait of shape optimization problems is that the space of designs is not a normed vector space. Therefore, tools to detect optima from analysis like distance, convergence, continuity and differentiation are not available. A way to furnish the set of shapes with these structures is the *perturbation of identity method* from [Murat and Simon, 1976a], or the *speed method* from [Sokolowski and Zolésio, 1992]. In these cases, the local shape variations are given as perturbations of the current domain from a function space which induces the desired properties. Then, the SOP resembles a standard optimal control problem over a function space instead of over a set of domains. Therefore, one can proceed to derive optimality criteria involving the gradients of cost function and PDE w.r.t. to the perturbations.

Shape calculus As in standard analysis, solutions to optimal control problems or shape optimization problems respectively are stationary points of the cost functional J, meaning that the gradient of J vanishes at optimal controls or domains. Shape calculus is the tool with which gradients w.r.t. domains can be defined, for instance by perturbation of identity. There are also other angles from which shape gradients can be viewed, e.g. from a Riemannian perspective [Schulz, 2014]. In the case of perturbation of identity this shape gradient is identified as the Fréchet-derivative of J at a perturbation.

Optimal control The SOP in (1.1) is an optimal control problem. However, optimal control theory is not directly applicable because of the aforementioned lack of vector space structures of the shape space. Shape optimization methods do use its concepts, though, to tackle SOPs and therefore knowledge from this area of research, for instance from [Tröltzsch, 2010] or [Hinze et al., 2009], provides helpful insight to the treatment of SOPs. A number of publications also deal with embedding SOPs into the standard control theoretic frame by transforming the shape optimization problem on a fixed reference domain and optimizing over transformations, i.e. functions, instead of shapes, e.g. [Eppler et al., 2007], [Ito et al., 2008] and [Brandenburg et al., 2009], and [Kiniger, 2015]. With these settings questions on existence, uniqueness, convergence and a priori error estimates can be answered. The transformation approach seems attractive for isogeometric analysis, as there a parameterization of the physical domain pulls the PDE back to a parameter domain anyway.

Discretization Usually, iterative, numerical optimization methods are employed to find the optimal perturbation or shape respectively. For that, both the function space for the PDEs and the space of perturbations have to be discretized to obtain a finite-dimensional problem. Basically, there are two approaches to discretize optimal control problems [Hinze et al., 2009] which is reflected in shape optimization: Either, one first uses shape calculus to derive the optimality system for the infinite-dimensional problem and then discretizes all function spaces and operators. This is the *optimize first–discretize then* approach. Or, in the second way *discretize first–optimize then*, one reverses the order of optimization and discretization which means the optimality system is derived for a finite-dimensional problem. Since there is quite a gap between the two communities using *optimize first* and *discretize first*, the question of their differences in isogeometric shape optimization arises.

Optimization methods To find a minimum numerically, a gradient-based optimization method is applied to a finite-dimensional nonlinear optimization problem resulting from either the *optimize first* or the *discretize first* approach. Hence, the shape gradient obtained by shape calculus serves two purposes, namely to give a first order optimality condition and furthermore, to indicate descent directions for the optimization routine. Several such black box solvers are applicable in shape optimization to receive update information based upon the gradients. However, a domain update from large deformations may lead to an infeasible mesh when a piece of boundary is moved into the inside of the domain. This is one of the practical bottlenecks of shape optimization, not only with IGA but also for classical approaches. The treatment is problem dependent and comprises techniques for instance from flow problems or r-adaptivity, [Budd et al., 2009].

Combining IGA and shape optimization The combination of IGA and SOPs is applied to a number of applications such as

- shells, [Kiendl et al., 2014],
- electrostatics, [Nguyen et al., 2012], [Bandara et al., 2015],
- fluid mechanics, [Nørtoft and Gravesen, 2013],

- solid mechanics, [Wall et al., 2008], [Qian, 2010], [Blanchard et al., 2013], [Fußeder et al., 2015], [Fußeder and Simeon, 2015],
- vibrating membranes, [Nguyen et al., 2011].

These publications show that IGA is suitable for shape optimization. Arguments for this combination are that B-spline and NURBS theory in CAGD puts a lot of effort into representing shapes exactly with just a finite number of points. However, often FEM destroys this effort by using piecewise linear approximations of the designs for analysis. In contrast, all occurring approximation spaces in IGA can be covered by one common description, namely B-splines or NURBS, without discarding information of the initial CAGD geometry model. Moreover, the set of admissible domains represented by B-splines/NURBS is larger than a space of polygons and regularity assumptions on the geometry can be more easily met. To the best of our knowledge, the *optimize first–discretize then* ansatz is only considered in [Blanchard et al., 2013] and [Bandara et al., 2015], where not the method itself is subject of investigation but its application to a particular SOP. Moreover, for [Bandara et al., 2015] an isogeometric *boundary element* method is used. This means that a more general investigation of the important *optimize first–discretize then* approach for IGA is still missing.

Contributions The aim of this thesis is to develop an abstract shape optimization framework in the *optimize first–discretize then* setting with a transformation approach which then is discretized by B-splines or NURBS and therefore comes natural to isogeometric analysis; one aspect is how to incorporate NURBS with variable weights in it. We compare this scheme to a *discretize first–optimize then* method to settle the question if they differ for isogeometric shape optimization and our class of problems. To complete the comprehensive view of IGA in shape optimization we present an algorithm which also takes into account the practical issues. With some examplary applications we fortify the developed theory.

We close this introduction with the structure of this thesis as follows.

Structure of the thesis

The two building blocks of shape optimization are linear elliptic partial differential equations and optimal control concepts. In **Chapter 2** we introduce the mathematical models for the state equations in this thesis together with their variational formulations. We transform them to a reference domain for IGA and for the abstract shape optimization framework. Important techniques and results from optimization with PDEs are summarized. Altogether this chapter provides the mathematical background for this thesis.

We then construct a continuous shape optimization framework in **Chapter 3**. In particular, we review the perturbation of identity method. This general framework is then considered under geometry transformations which ultimately aims at isogeometric discretization in the next chapter:

For the numerical treatment we discretize the control and state with isogeometric analysis. This means that **Chapter 4** combines discrete versions of Chapters 2 and 3. B-splines and NURBS are at its core and after a brief introduction we use them to obtain CAGD models for IGA. Subsequently, they also serve as test functions for the

Galerkin projection of the transported problem and for the shape optimization i.e. we search for optimal shapes also in this mutual B-spline/NURBS space.

Chapter 5 deals with practical aspects in the shape optimization process like the choice of optimization methods and mesh update strategies to avoid infeasible meshes. Moreover, we partly unravel the tight link of geometry, simulation, and optimization by using different B-spline spaces for each. This eventually also makes local adaptive refinement for the simulation possible.

Applications in **Chapter 6** finally illustrate the theory of isogeometric shape optimization. First, all previous computational aspects are summarized in an isogeometric shape optimization algorithm. Then secondly, we concentrate on particular problems, considering Poisson and linear elasticity state equations, to show the influence of discretization parameters and the decoupling of simulation and optimization meshes. In particular, local adaptive refinement for solving the state equation is realized. Moreover, we also treat rational B-spline optimization in the *optimize first* setting.

We conclude this thesis in **Chapter 7** where we summarize the results and also give an outlook on possible future steps.

2 Mathematical Modeling with PDEs

This chapter provides the foundation to formulate and solve shape optimization problems: A vital part of SOPs are partial differential equations whose solutions frequently enter the cost functionals in (1.1). For our SOPs, we consider linear elliptic equations of second order given by the state equation $e(\Omega, u) = 0$, which we introduce in this chapter. In particular, we formulate in **Section 2.2** the equations for our applications later on in Chapter 6. To find their numerical solution by either finite elements or isogeometric analysis we also express them in their variational form in **Section 2.3**. Therefore, these formulations are the basis for Galerkin discretization in Chapter 4.3. Since for all shapes in our SOP the state equation must be satisfied, it poses a constraint for the optimization problem. Thus we need the theory of optimization with PDEs, given in **Section 2.4**, to derive the shape optimization framework in Chapter 3.

However, we begin this chapter with **Section 2.1** of basic notations.

2.1 Preliminaries

In this section we fix the notation for the functional analysis background of this thesis. Particularly, we start with the notation for standard differential operators and quickly move on to a collection of important function spaces in which our SOPs are posed.

2.1.1 Basic definitions and notations

We begin with the usual abbreviation for total and partial derivatives for scalar functions $\phi(t)$, $\phi(x)$ and $\phi(x,t)$ where $t \in \mathbb{R}$ and $x \in \mathbb{R}^d$, $d > 0$, with components x_i for $i = 1, \dots, d$,

$$\mathrm{d}_t\phi := \frac{\mathrm{d}\phi}{\mathrm{d}t}, \quad \partial_t\phi := \frac{\partial\phi}{\partial t}, \quad \partial_i\phi := \frac{\partial\phi}{\partial x_i}, \quad \nabla\phi := (\partial_1\phi, \dots, \partial_d\phi)^\intercal \quad \text{etc.} \tag{2.1}$$

The derivative in a direction $\nu \in \mathbb{R}^d$ is denoted by $\partial_\nu\phi$ and is $\partial_\nu\phi = \nabla\phi \cdot \nu$ if ϕ is differentiable, where the dot notation $\cdot$ stands for the standard scalar product in $\mathbb{R}^d$. The differential operator D acts on vector valued functions $\phi\colon \mathbb{R}^d \to \mathbb{R}^m$ with $\phi(x) = (\phi_1, \dots, \phi_m)^\intercal$ and each $\phi_i := \phi_i(x)$ over $x \in \mathbb{R}^d$ like

(i) $m = 1$: $\mathrm{D}\phi = (\nabla\phi)^\intercal$ the transpose of the gradient,

(ii) $m \geq 1$: $\mathrm{D}\phi = \begin{pmatrix} \frac{\partial\phi_1}{\partial x_1} & \cdots & \frac{\partial\phi_1}{\partial x_d} \\ \vdots & & \vdots \\ \frac{\partial\phi_m}{\partial x_1} & \cdots & \frac{\partial\phi_m}{\partial x_d} \end{pmatrix} = \mathrm{J}_\phi$, the Jacobian,

(iii) $m = s \times d$: $\mathrm{D}\phi = \{\partial_1\phi, \dots, \partial_d\phi\}$. It shall act on a vector $v \in \mathbb{R}^d$ as a series of matrix-vector operations and yields a matrix of directional derivatives

$$\mathrm{D}\phi \cdot v = \Big(\partial_1\phi \cdot v, \dots, \partial_d\phi \cdot v\Big) \in \mathbb{R}^{s\times d}.$$

Usually we mean by $|.|$ the Euclidean norm in $\mathbb{R}^d$. For a metric space (X, d) with metric d we denote by $B_X(x, r) := \{y \in X\colon \mathrm{d}(x, y) < r\}$ the *open ball* of radius $r > 0$ around x.

2.1.2 Function space preliminaries

In this section we introduce the function spaces for linear elliptic PDEs of second order, in particular Sobolev spaces, the habitat of weak solutions. For that, we require the notion of weak derivatives and Lebesgue spaces. A basic reference is for instance [Adams and Fournier, 2003]. The smoothness of domains co-determines the regularity of weak solutions and typically the domains have to fulfill some regularity considerations, for instance having a boundary parameterization that is Lipschitz continuous. For the description of such domains and also for shape calculus later on, we introduce Hölder spaces.

Continuous functions

Differential operators $\partial^\alpha = \partial_1^{\alpha_1} \cdots \partial_d^{\alpha_d}$ are specified by multi-index $\alpha := (\alpha_1, \dots, \alpha_d)$.

Definition 2.1. On an open subset $\Omega \subset \mathbb{R}^d$ with boundary $\partial\Omega$ and closure $\bar{\Omega}$ all real-valued functions u on Ω which are continuous up to their m-th partial derivatives $\partial^\alpha u$ for $0 \le |\alpha| \le m$ form the *space of continuously differentiable functions* $\mathcal{C}^m(\Omega)$, with the special case $m = 0$ for *continuous functions* $\mathcal{C}(\Omega) := \mathcal{C}^0(\Omega)$.

For $m = \infty$ we get the space of *infinitely differentiable functions*:

Definition 2.2. $\mathcal{C}^\infty(\Omega) := \bigcap_{m=0}^\infty \mathcal{C}^m(\Omega)$, and $\mathcal{C}_0^m(\Omega)$ contains all functions u in $\mathcal{C}^m(\Omega)$ with compact support in Ω, i.e. if $\mathrm{supp}(u) := \overline{\{x \in \Omega\colon u(x) \neq 0\}}$ is contained in Ω and compact.

Sometimes we also make use of the notation $\mathcal{C}^k(\Omega; Y)$ for vector smooth functions from Ω to $Y \subset \mathbb{R}^m$ where $m \ge 1$. In a next step, we define the Hölder spaces $\mathcal{C}^{m,\lambda}(\bar{\Omega})$ with the help of the previously introduced space of continuous functions:

Definition 2.3. For $0 < \lambda \le 1$ a *Hölder space* is given by

$$\mathcal{C}^{m,\lambda}(\bar{\Omega}) := \{u \in \mathcal{C}^m(\bar{\Omega})\colon |\partial^\alpha u(x) - \partial^\alpha u(y)| \le C|x - y|^\lambda \,\forall x, y \in \Omega \text{ and } |\alpha| = m\}$$

for some constant C. For $m = 0$, $\lambda = 1$, $\mathcal{C}^{0,1}(\bar{\Omega})$ becomes the space of *Lipschitz continuous* functions.

The space of continuous functions with the supremum norm $\|u\|_\infty := \sup_{x\in\Omega}|u(x)|$ is a Banach space. We can also provide $\mathcal{C}(\bar{\Omega})$ with a scalar product $(u, v) := \int_\Omega uv \,\mathrm{d}x$. The induced norm $\|u\|_{L^2(\Omega)} := \sqrt{(u, u)}$ is, however, not equivalent to the infinity norm and therefore $\mathcal{C}(\bar{\Omega})$ is not complete under the L^2-norm, [Alt, 2012, p. 2 ff.]. Thus, not every Cauchy sequences must converge to elements in $\mathcal{C}(\bar{\Omega})$, and not every PDE solution we construct belongs to this space [Evans, 2010, p. 241]. As a completion of $\mathcal{C}(\bar{\Omega})$ under the L^2-norm we introduce next the Lebesgue spaces.

Lebesgue spaces

We now introduce the spaces of (Lebesgue-) integrable functions which have a norm and are complete: The *Lebesgue spaces* are defined as

$$L^p(\Omega) := \left\{ u\colon \Omega \to \bar{\mathbb{R}} \,\middle|\, \int_\Omega |u(x)|^p \,\mathrm{d}x < \infty \right\},\ 1 \le p < \infty,\ \bar{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\} \tag{2.2}$$

where we identify $u = v \Leftrightarrow \int_\Omega |u(x) - v(x)|^p \,\mathrm{d}x = 0$. Thus, its norm

$$\|u\|_{L^p(\Omega)} := \left(\int_\Omega |u(x)|^p \,\mathrm{d}x \right)^{1/p} \tag{2.3}$$

is well-defined. The vector space of *essentially bounded functions* on Ω is denoted by

$$L^\infty(\Omega) := \left\{ u\colon \Omega \to \bar{\mathbb{R}} \,\middle|\, \operatorname{ess\,sup} |u(x)| < \infty \right\} \text{ with its norm} \tag{2.4}$$

$$\|u\|_{L^\infty(\Omega)} := \operatorname*{ess\,sup}_{x\in\Omega} |u(x)| := \inf_M \sup_{x\in\Omega\setminus M} |u(x)| \text{ with null sets } M \subset \Omega\,. \tag{2.5}$$

Lebesgue spaces allow to introduce the notion of weak derivatives.

Definition 2.4. For $u \in L^1(\Omega)$ we call $w \in L^1(\Omega)$ a *weak derivative* of u if

$$\int_\Omega v w \,\mathrm{d}x = (-1)^{|\alpha|} \int_\Omega u\, \partial^\alpha v \,\mathrm{d}x \quad \forall v \in \mathcal{C}_0^\infty(\Omega)\,.$$

With this generalization of classical derivatives we finally obtain a wider class than $\mathcal{C}^k$-functions in which to solve our PDEs.

Sobolev spaces

Sobolev spaces are vitally important for the analysis of PDEs because with them we obtain their variational formulation and weak solution.

Definition 2.5. For a nonnegative integer k and $0 \le p \le \infty$ *Sobolev spaces*

$$W^{k,p}(\Omega) := \left\{ u\colon L^p(\Omega) \,\middle|\, \partial^\alpha u \in L^p(\Omega) \text{ for } 0 \le |\alpha| \le k \right\} \tag{2.6}$$

are linear subspaces of $L^p(\Omega)$ with weak derivatives ∂^α from Definition 2.4.

For $p \le \infty$, the functional

$$\|u\|_{W^{k,p}(\Omega)} := \left(\sum_{|\alpha|\le k} \|\partial^\alpha u\|_{L^p(\Omega)}^p \right)^{1/p}, \tag{2.7}$$

and for $p = \infty$, the functional

$$\|u\|_{W^{k,p}(\Omega)} := \max_{0\le|\alpha|\le k} \|\partial^\alpha u\|_{L^\infty(\Omega)}\,. \tag{2.8}$$

define norms in $W^{k,p}(\Omega)$. We mainly move in the Hilbert spaces $H^k(\Omega) := W^{k,2}(\Omega)$ in which functions and all their k-th weak derivatives are square-integrable. Furthermore, for vector valued problems like linear elasticity in Problems 2 and 3 in the next section we consider the product space $H^k(\Omega)^m$ which is the direct sum of $H^k(\Omega)$ spaces, and its norm

$$\|u\|_{H^k(\Omega)^m} = \left(\sum_{i=1}^{m} \|u_i\|^2_{H^k(\Omega)} \right)^{1/2} . \tag{2.9}$$

Analogously, define $W^{k,p}(\Omega)^m$. Frequently, the Sobolev semi-norms

$$|u|_{W^{k,p}(\Omega)} = \left(\sum_{|\alpha|=k} \|\partial^\alpha u\|^p_{L^p(\Omega)} \right)^{1/p} \tag{2.10}$$

are in use. Sobolev spaces with vanishing boundary values are given by

$$W_0^{k,p}(\Omega) := \{u \in W^{k,p}(\Omega) \colon \exists u_j \in \mathcal{C}_0^\infty(\Omega) \text{ s.t. } \|u - u_j\|_{W^{k,p}} \to 0 \text{ for } j \to \infty\}\,, \tag{2.11}$$

for instance in [Alt, 2012, 1.29, p. 68]. For smooth boundaries we can understand $W_0^{k,p}(\Omega)$ as functions $u \in W^{1,p}(\Omega)$ with $\partial^\alpha u = 0$ on $\partial\Omega$ for $|\alpha| \leq k-1$ in the sense of a trace operator. Again, we use $H_0^k(\Omega)$ to denote $W_0^{k,2}(\Omega)$. A definition of boundary smoothness is given next.

Classification of domains

It often is useful if the solution of the variational equation of a PDE has a higher regularity, for instance in *a posteriori* error estimation for FEM, and in shape optimization to represent a shape gradient simpler. The regularity depends among other factors on the smoothness of the domain over which the PDE is posed. We follow [Hinze et al., 2009, p. 19] to classify bounded domains.

Definition 2.6. Let Ω be a *domain*, i.e. a nonempty, open, bounded, connected set in $\mathbb{R}^d$ with boundary Γ. Further, let $m \geq 0$ or $m = \infty$ and $\lambda \in [0, 1]$. Then Ω is of class $\mathcal{C}^{m,\lambda}$ if for any $x \in \Gamma$ there exists $r > 0$, $\sigma \in \{-1, +1\}$, $l \in \{1, \ldots, d\}$ and a function $\gamma \in \mathcal{C}^{m,\lambda}(\mathbb{R}^{d-1})$ such that for the open ball $B_\Omega(x, r) := \{y \in \Omega \colon |x - y| < r\}$

$$\Omega \cap B_\Omega(x, r) = \{y \in B_\Omega(x, r) \colon \sigma y_l < \gamma(y_1, \ldots, \gamma_{l-1}, y_{l+1}, \ldots, \gamma_d)\} \text{ holds.}$$

For $\gamma \in \mathcal{C}^{0,1}(\mathbb{R}^{d-1})$ we also call Γ *Lipschitz boundary* and Ω *Lipschitz domain.*

This definition means that for every $x \in \Gamma$ there exists a neighborhood of x whose intersection with Γ is the graph of a $\mathcal{C}^{m,\lambda}$-function. For Lipschitz domains, the outward pointing unit normal $n \in \mathbb{R}^d$ exists almost everywhere on Γ.

We now have the requisites to formulate our model PDE problems.

2.2 Model Problems

In this section we introduce the mathematical description for physical phenomena and engineering applications with PDEs which we consider for the rest of this thesis as working models. In particular, we restrict ourselves to linear elliptic partial differential equations of second order and start by giving a general notation. Subsequently, we narrow them down to Poisson's equation and linear elasticity.

Definition 2.7. A general *linear elliptic partial differential operator of second order*, [Evans, 2010, p. 295], acting on a function $u\colon \mathbb{R}^d \to \mathbb{R}$ is denoted by

$$\mathrm{L}u = -\sum_{i,j=1}^{d} a_{i,j}(x)\partial_{ij}u + \sum_{i=1}^{d} b_i(x)\partial_i u + c(x)u \tag{2.12}$$

with coefficients $a_{i,j}$, b_i, c_i. We assume that L is *uniformly elliptic*: There exists a constant $\alpha > 0$ such that

$$\sum_{i,j=1}^{d} a_{i,j}\xi_i\xi_j > \alpha\|\xi\|^2 \quad \text{for a.e. } x \in \Omega,\ \forall \xi \in \mathbb{R}^d\,. \tag{2.13}$$

The notation can also be extended to vector-valued solutions $u\colon \mathbb{R}^d \to \mathbb{R}^m$ with partial derivatives of the components

$$\partial_{ij}u_k,\ \partial_i u_k,\ \text{for } i,j = 1,\dots,d,\ \text{and } k = 1,\dots,m\,. \tag{2.14}$$

A boundary value problem is thus given in the strong form by

$$\mathrm{L}u = f \text{ in an open bounded set } \Omega \tag{2.15}$$

together with appropriate boundary conditions on the boundary $\Gamma = \partial\Omega$

$$u = g_D \text{ on a Dirichlet boundary } \Gamma_D \subset \partial\Omega, \tag{2.16}$$

$$\partial_n u = g_N \text{ on a Neumann boundary } \Gamma_N \subset \partial\Omega \tag{2.17}$$

where $\Gamma_D \cap \Gamma_N = \emptyset$ and $\Gamma_D \cup \Gamma_N = \Gamma$. The formulation (2.15)–(2.17) covers our model problem for $m = 1$, the *Poisson equation*. The latter describes several physical processes, e.g. steady-state heat conduction, electrostatics, or deformation of a thin elastic membrane, see also [Atkinson and Han, 2001, pp. 328].

Problem 1 (Poisson).
Let $\mathrm{L}u = \operatorname{div}(\nabla u) = \sum_{i=1}^{d} \partial_{ii}u =: \Delta u$ *be the Laplace operator acting on* u*, then the Poisson equation is given by*

$$\begin{cases} -\Delta u &= f & \text{in } \Omega\,, \\ u &= g_D & \text{on } \Gamma_D\,, \\ \partial_n u &= g_N & \text{on } \Gamma_N\,. \end{cases} \tag{2.18}$$

On the boundary $\partial\Omega$ *may govern Dirichlet conditions on* Γ_D *or Neumann conditions on* Γ_N*. We call* u *a* classical solution *if it satisfies* (2.18) *pointwise and is an element of* $\mathcal{C}^2(\Omega) \cap \mathcal{C}^1(\bar{\Omega})$*. In case of* $\Gamma = \Gamma_D$*, a classical solution is a member of* $\mathcal{C}^2(\Omega) \cap \mathcal{C}(\bar{\Omega})$*.*

Contrary to Problem 1, the linear elasticity equation is an example of a vector valued boundary value problem (2.15)–(2.17) of dimension $m = 2$ or 3. It models deformations of an elastic body of St. Venant-Kirchhoff material under the influence of external forces like traction or gravity, [Gekeler, 2010, p. 417].

Problem 2 (Linear Elasticity).
For d-dimensional bodies $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, let $\phi\colon \Omega \times [0,\tau] \to \mathbb{R}^d$ denote the deformation *of the reference domain Ω which maps $(x,t) \mapsto x(t) := x + u(x,t)$. We refer to $x \in \Omega$ as* material *or* Eulerian *coordinates, whereas $x(t)$ are* space *or* Lagrangian *coordinates. This deformation is defined in terms of the* displacement *$u\colon \Omega \times [0,\tau] \to \mathbb{R}^d$. In the* stationary case*, the displacement is characterized by the linear elasticity equation*

$$\begin{cases} -\operatorname{div}\sigma(u) &= f & \text{in } \Omega\,, \\ u &= g_D & \text{on } \Gamma_D\,, \\ \sigma(u)n &= g_N & \text{on } \Gamma_N\,, \\ \sigma(u)n &= 0 & \text{on } \Gamma \end{cases} \tag{2.19}$$

with $\partial\Omega = \Gamma_N \cup \Gamma_D \cup \Gamma$. This strong form arises from a linear material law for the strain

$$\varepsilon(u) = \frac{1}{2}(\nabla u + \nabla u^{\mathsf{T}}) \tag{2.20}$$

and Hooke's law for the strain-stress relationship

$$\sigma(u) = 2\mu\varepsilon(u) + \lambda(\operatorname{tr}\varepsilon(u))I \tag{2.21}$$

where tr *denotes the trace operator. The Lamé parameters*

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}\,, \quad \mu = \frac{E}{2(1+\nu)} \tag{2.22}$$

are related to constants ν and E. Those are the material properties, Young's modulus E, measuring axial stiffness, and Poisson number ν, measuring lateral contraction, [Atkinson and Han, 2001, Ch. 8.5]. In formulation (2.19)*, σ is a stress tensor for which the i-th component is $(\operatorname{div}\sigma)_i = \sum_{j=1}^m \partial_j \sigma_{ij}$. Due to symmetry reasons, we often also use the Voigt notation. That is, instead of considering the $m \times m$ sized tensors σ and ε we use a vector notation, here for $m = 3 = d$, $\underline{\sigma} = (\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23})^{\mathsf{T}}$ and $\underline{\varepsilon} = (\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, 2\varepsilon_{12}, 2\varepsilon_{13}, 2\varepsilon_{23})^{\mathsf{T}}$,* (2.21) *yielding $\underline{\sigma} = C\underline{\varepsilon}$ with*

$$C = \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}(1-2\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}(1-2\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2}(1-2\nu) \end{pmatrix}.$$

A variant of linear elasticity in two dimensions is given for very thin plates in Problem 3 where we only view the cross section Ω in two dimensions:

Problem 3 (Plane Stress).
In a plane stress setting, forces can only act into in the plane of the cross section and not into a third direction, [Gekeler, 2010, p. 420]. For $m = 2 = d$, we consider the strong form (2.19) *with*

$$\lambda = \frac{\nu E}{1-\nu^2}, \quad \mu = \frac{E}{2(1+\nu)} \text{ and } C = \frac{E}{1-\nu^2}\begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1}{2}(1-\nu) \end{pmatrix}. \tag{2.23}$$

Remark 2.8. For ease of notation we mostly assume homogeneous Dirichlet boundary conditions in Problems 1–3, i.e. $g_D = 0$. For a formulation with non-homogeneous Dirichlet boundary conditions we refer to standard literature such as the above or [Atkinson and Han, 2001].

Operator notation Finally, we say that $\mathrm{e}(\Omega, u) = 0$ means that the boundary value problem (2.15) is fulfilled on Ω, i.e. that $Lu - f = 0$ on Ω, $u = 0$ on Γ_D and $\partial_n u - g_N = 0$ on Γ_N. The domain and range spaces will be specified later. For the moment we assume that Ω is fixed and $\mathcal{U}$, $\mathcal{Z}$ are real Banach spaces of functions defined over Ω. Then let $u \in \mathcal{U}$ and $\mathrm{e}(\Omega, \cdot)\colon \mathcal{U} \to \mathcal{Z}$. Note, that in this thesis we consider only stationary problems, although this operator notation would allow time dependend problems as well.

To solve our problems numerically, the starting point is their variational form which we introduce next.

2.3 Variational Formulation of Elliptic PDEs

Seemingly, a $\mathcal{C}^2$-function space seems attractive for second order PDEs from Section 2.2 due to providing second derivatives. However, it usually is easier to find solutions and prove their existence and uniqueness in a space with less smoothness. For that we introduce weak solutions in appropriate Sobolev spaces and state existence and uniqueness results following from the Lax-Milgram Lemma. In particular, such weak solutions solve a variational formulation of the PDE. In that form, it needs not satisfy the PDE pointwise anymore but weakly in an integral over multiplications with test functions. The variational equation is also the starting point for numerical discretization by finite element methods or isogeometric analysis in Section 4.3. In IGA, however, we first reformulate the variational equation over a parameter domain. There, we have a transformation map between the parameter domain and the domain of the PDE which is used in a change of variables in the integrations. Similarly, also shape calculus uses a change of variables in integrals to obtain derivatives w.r.t. domains in Chapter 3.

We proceed by first introducing the weak form for the PDEs given by $\mathrm{e}(\Omega, u) = 0$ from Section 2.2. Second, we state the Lax-Milgram Lemma, and third, a change of variables for variational equations.

2.3.1 Weak formulation of PDEs

In this section we obtain the variational formulation of PDEs that then allow us to make statements about existence and uniqueness of solutions in the next Section 2.3.2.

The weak forms also serve as starting point for the discretization process in Chapter 4.

We say that u in a Hilbert space $\mathcal{V}$ is a *weak solution* of the PDE $\mathrm{e}(\Omega, u) = 0$ and thus of (2.15)–(2.17), if

$$u \in \mathcal{V}\colon a(u,v) = l(v) \quad \forall v \in \mathcal{V}\,, \tag{2.24}$$

with *bilinear form* $a\colon \mathcal{V} \times \mathcal{V} \to \mathbb{R}$

$$a(u,v) := \int_\Omega \left(\sum_{i,j=1}^{d} a_{i,j} \partial_i u \partial_j v + \sum_{i=1}^{d} b_i\, \partial_i u v + cuv \right) \mathrm{d}\Omega \tag{2.25}$$

and *linear form*

$$l(v) := \int_\Omega f v \,\mathrm{d}\Omega + \int_{\Gamma_N} g_N v \,\mathrm{d}\Gamma\,. \tag{2.26}$$

The coefficients $a_{i,j}$, b_i, and c correspond to those of L. For vector valued functions we again consider components $a_{ik,j\ell}\partial_i u_k \partial_j v_\ell$ etc, as in (2.14) as well as scalar products $f \cdot v$ and $g_N \cdot v$ in (2.26). Homogeneous Dirichlet boundary conditions are realized directly in the test function space $\mathcal{V}$.

We make some further assumptions and introduce properties of the bilinear form that we need for showing that the variational equation has a solution.

Assumption 2.9. Assume that the coefficients $a_{i,j}$, b_i, and c are bounded in $L^\infty(\Omega)$. Let $\partial\Omega = \Gamma_D \cup \Gamma_N$. The function space $\mathcal{V}$ usually is the Hilbert space $\mathcal{V} := H^1_{\Gamma_D}(\Omega)^m := \{v = (v_1, \dots, v_m) \in H^1(\Omega)^m\colon v|_{\Gamma_D} = 0\}$. We presume sufficient regularity of the domain, e.g., Ω is polygonal and convex, or a Lipschitz domain (Definition 2.6), or has a $\mathcal{C}^2$-boundary. Finally, let $f \in L^2(\Omega)^m$ and $g_N \in L^2(\Gamma_N)^m$.

Definition 2.10. A bilinear form $a\colon \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ on Hilbert spaces $\mathcal{V}$ with norm $\|\cdot\|$ is

- *bounded* if there exists a constant $M < \infty$ such that $a(u,v) \le M\|u\|\|v\|$ $\forall u, v \in \mathcal{V}$.
- *$\mathcal{V}$-elliptic* if there exists a constant $\alpha > 0$ such that $a(v,v) \ge \alpha\|v\|$ $\forall v \in \mathcal{V}$.
- *symmetric* if $a(u,v) = a(v,u)$ $\forall u, v \in \mathcal{V}$.

Remark 2.11. For $b_i = 0$, $i = 1, \dots, d$, and $c = 0$ it follows directly that the bilinear form in (2.25) is bounded and $\mathcal{V}$-elliptic under Assumption 2.9. Else, additional restrictions have to be imposed on the coefficients, see [Atkinson and Han, 2001, p. 345].

From classical to variational formulation We obtain the weak form (2.24) of a problem (2.15) by multiplying the strong form with test functions from $\mathcal{V}$ and using integration by parts.

Operator notation Commonly in optimal control and shape optimization, the testing is expressed in an operator notation. Consider the weak solution $u \in \mathcal{V}$ of $\mathrm{e}(\Omega, u) = 0$ where the domain Ω is fixed. Then we can say that $\mathrm{e}(\Omega, u)\colon \mathcal{V} \to \mathbb{R}$ defines

for any $u \in \mathcal{V}$ a linear and continuous functional and by testing with functions from $\mathcal{V}$ obtain the variational form (2.24) in operator notation

$$\langle \mathrm{e}(\Omega, u), v \rangle_{\mathcal{V}^*, \mathcal{V}} = 0 \quad \forall v \in \mathcal{V} \,. \tag{2.27}$$

To see this we write $\mathrm{e}(\Omega, u) = l(\cdot) - a(u, \cdot)$ and note that $l(\cdot)$ and $a(u, \cdot)$ are linear and continuous by definition. The notation can be found e.g. in [Hinze et al., 2009].

Our next step is to make sure under which conditions we have unique weak solutions.

2.3.2 Existence of solutions

The Lax-Milgram Lemma is central to show if elliptic partial differential equations can be solved uniquely.

Theorem 2.12 (Lax-Milgram Lemma).
For a Hilbert space $\mathcal{V}$, a bounded, $\mathcal{V}$-elliptic bilinear form $a \colon \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ and a linear functional $l \in \mathcal{V}^*$ there exists a unique solution of

$$u \in \mathcal{V} \colon a(u, v) = l(v) \quad \forall v \in \mathcal{V} \,. \tag{2.28}$$

Proof. See [Atkinson and Han, 2001, Th. 8.3.4, p. 336]. □

In case of a symmetric bilinear form the variational equation (2.28) is equivalent to a minimizing problem:

Theorem 2.13.
Assume that all assumptions of Lax-Milgram Theorem 2.12 are fulfilled, Y is a non-empty, closed, convex subset of $\mathcal{V}$, and a is symmetric. Then there exists a unique $u \in Y$ that is a minimizer of

$$J(v) := \frac{1}{2} a(v, v) - l(v) \,, \quad \forall v \in Y \,, \tag{2.29}$$

i.e. $J(u) = \inf_{v \in Y} J(v)$.

Proof. See [Atkinson and Han, 2001, p. 336]. □

With Theorem 2.12 and Remark 2.11 we directly conclude that Problem 1 with homogeneous Dirichlet boundary conditions $\partial\Omega = \Gamma_D$ and $g_D = 0$ has under Assumption 2.9 a unique solution. In case of Neumann boundary conditions additional assumptions are necessary. For linear elasticity Problems 2 and 3 the $\mathcal{V}$-ellipticity of the bilinear form is verified with Korn's inequality and thus also uniquely solvable, see for instance [Atkinson and Han, 2001, Th. 8.5.1, p. 352].

We next consider the existence result for equations under transformations.

2.3.3 Change of variables

The two topics of this thesis, IGA and shape optimization, use a change of variables in the integration terms to pose their respective problems over a reference domain instead of a physical domain:

- In IGA there is a transformation of a parameter domain $\hat{\Omega}$ to the domain Ω over which a PDE is posed. With the help of this parameterization, the PDE is solved over $\hat{\Omega}$ instead and afterwards transported to the solution over Ω.
- In shape optimization, variations of a domain Ω are given by transforming it to slightly perturbed domains. The shapes appear as domains of integration and to move them to the integrand, a change of variables is applied.

Since a change of variables is so central to the two topics of this thesis, we take a closer look when it holds for our integrals and PDEs.

Remark 2.14. Before starting on the change of variables we make a concession to the suggestive use of terms *pull back* and *push forward*. It is common in IGA, see for instance [Bazilevs et al., 2006], that the composition with a map $G^{-1}\colon \Omega \to \hat{\Omega}$ between two manifolds itself as well as the push-forward of the Jacobian are both called the push-forward, and its inverse G pull-back, respectively. Therefore, an isogeometric map G between manifolds can pull back something to the parameter domain $\hat{\Omega}$ or push it forward onto the physical domain Ω. We use this vocabulary intuitively for other transformations as well.

We start with a corollary of a more general theorem in [Varberg, 1971] concerning the applicability of the transformation formula for integrals.

Lemma 2.15. *Let $T\colon U \to \mathbb{R}^d$ be differentiable (or Lipschitz continuous) on an open set $U \subset \mathbb{R}^d$. If T is one-to-one on a subset of U whose complement in U has measure 0, then the change of variables formula*

$$\int_{T(\Omega_0)} \phi(x)\,\mathrm{d}x = \int_{\Omega_0} \phi \circ T(\hat{x}) |\det \mathrm{J}_T(\hat{x})|\,\mathrm{d}\hat{x} \tag{2.30}$$

where J_T is the Jacobian of T, holds for each measurable subset $\Omega_0 \subset U$ and $\phi \in L^1(T(\Omega_0))$.

We next answer the question of change of variables in integrals for two important transformation classes, diffeomorphisms and bi-Lipschitz maps.

Transformations with diffeomorphisms Let T denote a $\mathcal{C}^1$-diffeomorphism from a domain $\Omega_0 \subset \mathbb{R}^d$ onto the domain $\Omega \subset \mathbb{R}^d$ with bounded derivatives, i.e. for $k > 0$ there are real constants c, C such that

$$T\colon \Omega_0 \to \Omega, \text{ where } T \in \mathcal{C}^k(\Omega_0),\ T^{-1} \in \mathcal{C}^k(\Omega) \text{ and } 0 < c \leq |\det \mathrm{J}_T| \leq C \tag{2.31}$$

with Jacobian $\mathrm{J}_T := (\partial T_i / \partial \hat{x}_j)_{ij}$. Then a transformation between Sobolev spaces holds:

Theorem 2.16.
Let T satisfy (2.31) for open domains Ω_0 and Ω, $1 \leq p \leq \infty$, then T transforms the Sobolev space $W^{s,p}(\Omega_0)$ boundedly onto $W^{s,p}(\Omega)$ and has a bounded inverse, i.e. if $\phi \in W^{s,p}(\Omega)$ then $\phi \circ T \in W^{s,p}(\Omega_0)$. For $s = 1$ the chain rule for the weak derivatives of $\phi \circ T$ is

$$\partial_i(\phi \circ T) = \sum_{j=1}^{d} (\partial_j \phi) \circ T\, \partial_i G_j\,.$$

Proof. A proof can be found in [Adams and Fournier, 2003, Theorem 3.41, p. 78]. The chain rule is proved in [Alt, 2012, Ch. 2, Th. 2.26]. □

This result also holds for Sobolev spaces that include Dirichlet boundary conditions.

Transformations with bi-Lipschitz homeomorphisms Let T denote a *bi-Lipschitz* one-to-one mapping from a domain $\Omega_0 \subset \mathbb{R}^d$ onto the domain $\Omega \subset \mathbb{R}^d$, i.e.

$$T\colon \Omega_0 \to \Omega, \text{ where } T \in \mathcal{C}^{0,1}(\bar{\Omega}_0), \text{ and } T^{-1} \in \mathcal{C}^{0,1}(\bar{\Omega})\,. \tag{2.32}$$

The next theorem mimics Theorem 2.16 for a change of variables in integrations with these transformations.

Theorem 2.17.
Let T satisfy (2.32) for bounded, open sets Ω_0, Ω and $1 \leq p < \infty$ then T transforms $L^p(\Omega_0)$ onto $L^p(\Omega)$ boundedly, i.e. if $\phi \in L^p(\Omega)$ then $\phi \circ T \in L^p(\Omega_0)$. This also holds for ϕ in Sobolev spaces $W^{1,p}(\Omega)$.

Proof. The results for the bounded transformation for L^p and $W^{1,p}$ can be found as Lemma 3.1 and 3.2 in [Nečas, 2012, p. 60]. We only note here that the line of argumentation follows not the one of Theorem 2.16. □

Application to isogeometric analysis and shape optimization On the one hand, we have in IGA a transformation G between a parameter domain $\hat{\Omega}$ and the domain over which the PDE is posed, Ω. The parameterization G of Ω in IGA is exploited to pull back the variational equation of the PDE to the parameter domain and solve it there. In particular, also the function spaces of the variational form are transformed. Afterwards, the solution from the parameter setting is pushed forward to obtain the solution of the original problem. On the other hand, in shape optimization a domain Ω is perturbed or changed by mapping it to a new domain $\tau(\Omega)$. Shape sensitivities are obtained by considering again the pull back integrals to Ω by a change of variables. Then, the perturbation occurs in the integral kernel and can be differentiated.

In the following, we therefore show that under typical transformations from isogeometric analysis or shape optimization a change of variables is applicable and the two problems –original and transformed– are equivalent. We formulate the transformed variational equation.

Corollary 2.18. *Let T be a transformation that satisfies (2.31). For $\hat{u} := u \circ T \in \hat{\mathcal{V}} := H^1_{\hat{\Gamma}_D}(\Omega_0)^m$ a change of basis yields equivalent weak formulations (2.24) for the state equation, the PDE $\mathrm{e}(\Omega, u) = 0$:*

$$u \in \mathcal{V}\colon a(u,v) = l(v) \quad \forall v \in \mathcal{V} \tag{2.33}$$

$$\Leftrightarrow \quad \hat{u} \in \hat{\mathcal{V}}\colon \hat{a}(T)(\hat{u},\hat{v}) = \hat{l}(T)(\hat{v}) \quad \forall \hat{v} \in \hat{\mathcal{V}}\,, \tag{2.34}$$

where the transformed bilinear and linear forms are given by

$$\hat{a}(T)(\hat{u},\hat{v}) := \int_{\hat{\Omega}_0} \left(\sum_{i,j=1}^{d} \sum_{k,\ell=1}^{m} a_{ik,j\ell} \circ T (\mathrm{D}\hat{u}\,\mathrm{J}_T)_{k,i} (\mathrm{D}\hat{v}\,\mathrm{J}_T)_{\ell,j} + \right. \tag{2.35}$$

$$\left. \sum_{k,\ell=1}^{m} b_{k\ell} \circ T \hat{u}_k \hat{v}_\ell + c \circ T \hat{u} \cdot \hat{v} \right) |\det \mathrm{J}_T| \,\mathrm{d}\hat{\Omega}_0 \,, \tag{2.36}$$

$$\hat{l}(T)(\hat{v}) := \int_{\hat{\Omega}_0} f \circ T \cdot \hat{v} |\det \mathrm{J}_T| \,\mathrm{d}\hat{\Omega}_0 + \int_{\hat{\Gamma}_N} g_N \circ T \cdot \hat{v} |\mathrm{J}_T^{-\top} \hat{n}| |\det \mathrm{J}_T| \,\mathrm{d}\hat{\Gamma} \tag{2.37}$$

with outer normal $\hat{n}$ to the boundary $\hat{\Gamma}_N = T^{-1}(\Gamma_N)$ in the reference domain.

Proof. As a consequence from Lemma 2.15 for $\phi \in L^1(\Omega)$ the change of variables (2.30) holds for T

$$\int_{\Omega} \phi(x) \,\mathrm{d}x = \int_{\hat{\Omega}} (\phi \circ T)(\hat{x}) |\det \mathrm{J}_T(\hat{x})| \,\mathrm{d}\hat{\Omega} \tag{2.38}$$

and therefore for the forms $a(\cdot,\cdot)$ and $l(\cdot)$. Theorem 2.16 ensures that the coefficients $a_{ik,j\ell}$ and $b_{k\ell}$ are still bounded. The chain rule yields for a composite function $\hat{u} = u \circ T$

$$\mathrm{D}\hat{u} = \big((\mathrm{D}u) \circ T\big)\,\mathrm{D}T = ((\mathrm{D}u) \circ T)\,\mathrm{J}_T \,. \tag{2.39}$$

with differential operator D and Jacobian J_G as in Theorem 2.16, see also Appendix A for details to differentiation in multi-dimensions. Likewise, this theorem ensures then that if a solution $\hat{u}$ of the transported problem exists then $\hat{u} = u \circ T$. By assumption, the Jacobian is bounded, therefore the operators of the transformed bilinear and linear form are still bounded and $\hat{\mathcal{V}}$-elliptic. Thus, the transformed problem (2.34) has a unique solution by Lax-Milgram, too. □

Remark 2.19. Above theorem holds true also for bi-Lipschitz transformations (2.32) since their derivatives are bounded by the Lipschitz condition and Theorem 2.17 substitutes for Theorem 2.16 in the proof.

We comment on our notation:

Remark 2.20. The notation $a(u,v)$ versus $\hat{a}(T)(\hat{u},\hat{v})$ emphasizes that in the variational form (2.33), the function spaces $\mathcal{V} = H^1_{\Gamma_D}(\Omega)^m$ depend on $\Omega = T(\hat{\Omega})$ and thus on T in contrast to (2.34) where the dependency on T is moved to the operators in the bilinear and linear form.

In isogeometric analysis the reference domain is called parameter domain $\hat{\Omega}$ and is the unit hypercube in $\mathbb{R}^d$ for which the outer unit normal $\hat{n}$ is just a standard unit vector save for different signs. We from now on use the convention, that a hat $\hat{}$ indicates that something is defined on the parameter domain $\hat{\Omega}$. The isogeometric transformation is here denoted by a geometry function G which satisfies either (2.31) or (2.32).

We illustrate an isogeometric transformation in the following example.

Example 2.1. Let $\mathrm{e}(\Omega, u) = 0$ denote Poisson's equation with variational form

$$u \in H_0^1(\Omega)\colon \int_\Omega \nabla u \nabla v \,\mathrm{d}x = \int_\Omega f v \,\mathrm{d}x \quad \forall v \in H_0^1(\Omega)\,. \tag{2.40}$$

Because of a change of variables, the chain rule and bounded derivatives of G, equation (2.40) is equivalent to $\hat{u} = u \circ G \in H_0^1(\hat{\Omega})\colon$

$$\int_{\hat{\Omega}} \nabla \hat{u}\, \mathrm{J}_G^{-1}\, \mathrm{J}_G^{-\mathsf{T}} \nabla \hat{v} |\det \mathrm{J}_G| \,\mathrm{d}\hat{x} = \int_{\hat{\Omega}} f \circ G \hat{v} |\det \mathrm{J}_G| \,\mathrm{d}\hat{x} \quad \forall \hat{v} \in H_0^1(\hat{\Omega})\,. \tag{2.41}$$

To sum up so far, we have introduced the model state equations for our SOPs in the previous Section 2.2. Then, in this section, we derived their weak formulation and, important for IGA discretization as well as for shape calculus, the change of variables in integrals for these variational equations was reviewed. Since shape optimization is a special case of optimization with PDEs and of optimal control in particular, we continue with some of its main concepts which occur in shape optimization later.

2.4 Optimization with PDEs

Shape optimization problems are a special kind of optimal control problems for which we need techniques from optimization with PDEs. To handle SOPs they are brought from the general form in (1.1) to an optimal control frame, for instance with the perturbation of identity method. In Chapter 3 we also go that way and argue from an optimal control point of view the existence and optimality criteria for such problems. Therefore, this chapter is a road map for the abstract shape optimization treatment and provides notation, concepts and results for later.

After we introduce nonlinear optimal control problems, we briefly study under which conditions an existence of optimal controls is guaranteed. Then we state first order necessary optimality conditions and show how to derive them. These problems are posed in infinite-dimensional vector spaces, just like the shape optimization problems, therefore, the methodologies reflect this infinite nature. The differentiation in function spaces makes it necessary to draw to Fréchet-derivatives for which we state the notation and some useful results in Appendix B. Our exposition here follows primarily [Hinze et al., 2009].

2.4.1 Optimal Control Problems

Let the *state* be the solution u of the *state equation* $\mathrm{e}(q, u) = 0$ which denotes a PDE depending on *controls* q from the control space $\mathcal{Q}$. Assume that u is from a suitable function space $\mathcal{U}$, for instance for the following equation

$$\begin{cases} -\operatorname{div}(q \nabla u) &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \Gamma_D = \partial\Omega\,. \end{cases} \tag{2.42}$$

Furthermore, let $J\colon \mathcal{Q} \times \mathcal{U} \to \mathbb{R}$ define an *objective* or *cost functional*, e.g.

$$J(q, u) = \int_\Omega f(x) u(x) \,\mathrm{d}x \tag{2.43}$$

where the state u depends on the control $u(x) = u(q)(x)$. A general *optimal control problem* then has the structure

$$\min_{(q,u)\in\mathcal{Q}\times\mathcal{U}} J(q,u) \quad \text{s.t.} \quad \mathrm{e}(q,u) = 0\,, \text{ and } q \in \mathcal{Q}_{\mathrm{ad}} \subset \mathcal{Q} \tag{2.44}$$

see for instance [Hinze et al., 2009, p. 2]. In $\mathcal{Q}_{\mathrm{ad}}$ additional restrictions on the control can be made, an example follows shortly. To stay in the context of shape optimization, consider a sizing problem from [Hinze et al., 2009, p. 9].

Example 2.2. A very thin elastic membrane spanned over the domain $\Omega \subset \mathbb{R}^2$ is clamped at its boundary. Given a vertical force distribution $f\colon \Omega \to \mathbb{R}$ acting from below, the displacement is denoted by $u\colon \Omega \to \mathbb{R}$ and given by (2.42). The design goal of this SOP is to find an optimal thickness q such that the membrane is as stiff as possible, which means that the compliance (2.43) is minimized. Furthermore, the thickness is restricted within $a(x) \le q \le b(x)$ for $x \in \Omega$, and we enforce a volume constraint $\int_\Omega q(x)\,\mathrm{d}x \le V_0$ for a constant $V_0 \ge 0$.

In this form it is a "usual" optimal control problem and does not require special treatment from shape optimization like transformation to a reference domain because in its variational equation the space $\mathcal{V}$ is independent of q.

Since (2.44) is posed in infinite-dimensional spaces some concepts like compactness and convergence have to be reconsidered, see Appendix B.2 for details. With this in mind existence of optimal controls for problem (2.44) can be shown.

2.4.2 Existence of Optimal Controls

Consider the general problem (2.44). We assume that the state equation allows a unique solution for each control which is noted by the following operator.

Definition 2.21. A *control-to-state operator* is defined by

$$\mathrm{S}\colon \mathcal{Q} \to \mathcal{U}, \quad q \mapsto u(q)$$

if for all $q \in \mathcal{Q}$ the state equation $\mathrm{e}(q,u) = 0$ admits a unique solution u.

The existence of such an operator is typically shown with the Lax-Milgram Lemma 2.12. We denote the *admissible* or *feasible set* by

$$\mathcal{W}_{\mathrm{ad}} = \{(q,u) \in \mathcal{Q}\times\mathcal{U}\colon q \in \mathcal{Q}_{\mathrm{ad}} \text{ and } \mathrm{e}(q,u) = 0\} \tag{2.45}$$

and say that (2.44) has a *global solution* $(q^\star, u^\star)$ if

$$(q^\star,u^\star) \in \mathcal{W}_{\mathrm{ad}} \text{ and } J(q^\star,u^\star) \le J(q,u) \quad \forall (q,u) \in \mathcal{W}_{\mathrm{ad}}\,. \tag{2.46}$$

Under some assumptions such an optimal pair exists.

Theorem 2.22.
Let $\mathcal{Z}$, $\mathcal{U}$, $\mathcal{Q}$ be Banach spaces, $\mathcal{Q}$ and $\mathcal{U}$ are reflexive (definition in Appendix B.2.2, (vi)). Consider the optimal control problem (2.44) *where $J\colon \mathcal{Q}\times\mathcal{U} \to \mathbb{R}$ and $\mathrm{e}\colon \mathcal{Q}\times\mathcal{U} \to \mathcal{Z}$ are continuous. Under the assumptions*

(A1) $\mathcal{Q}_{ad} \subset \mathcal{Q}$ is convex, bounded and closed,

(A2) $\mathcal{U}$ is convex and closed such that (2.44) *has a feasible point,*
(A3) the state equation $\mathrm{e}(q,u)=0$ *has a bounded solution operator* $\mathrm{S}\colon q \mapsto u(q)$,
(A4) $(q,u) \in \mathcal{Q}\times\mathcal{U} \mapsto \mathrm{e}(q,u) \in \mathcal{Z}$ *is continuous under weak convergence,*
(A5) J is sequentially weakly lower semicontinuous (Definition B.11),
the optimal control problem (2.44) *has an optimal solution in* $\mathcal{Q}_{ad}\times\mathcal{U}$.

Proof. We state and comment on [Hinze et al., 2009, Theorem 1.48, p. 55]. We need here results of weak convergence which are stated in the Appendix B.2.4.

From *(A2)* we know $\mathcal{W}_{\mathrm{ad}} \neq \emptyset$ and

$$J^\star := \inf_{(q,u)\in\mathcal{W}_{\mathrm{ad}}} J(q,u) \text{ exists.}$$

There is a minimizing sequence $(q_n,u_n) \subset \mathcal{W}_{\mathrm{ad}}$

$$\lim_{n\to\infty} J(q_n,u_n) = J^\star\,.$$

Because of *(A1)* the sequence (q_n) is bounded; because of *(A3)* also (u_n) is bounded. In infinite spaces, the Weierstraß theorem to obtain a *convergent subsequence* is not applicable. However, by *(A1)* $\mathcal{Q}_{\mathrm{ad}}$ is weakly sequentially compact, and by *(A2)* $\mathcal{U}$ is weakly sequentially closed. Therefore we can extract a *weakly convergent subsequence*

$$(q_{n_k},u_{n_k}) \rightharpoonup (q^\star,u^\star) \in \mathcal{Q}\times\mathcal{U} \text{ for } k\to\infty\,.$$

Since $\mathcal{Q}_{\mathrm{ad}}\times\mathcal{U}$ is closed and convex by assumption, $\mathcal{W}_{\mathrm{ad}}$ is closed and convex. From *(A4)* it follows that $\mathcal{W}_{\mathrm{ad}}$ is sequentially weakly closed, hence $(q^\star,u^\star)\in\mathcal{W}_{\mathrm{ad}}$. For a weakly lower semicontinuous J in a reflexive Banach space, *(A5)*, it holds

$$(q_{n_k},u_{n_k}) \rightharpoonup (q^\star,u^\star) \Rightarrow \liminf_{k\to\infty} J(q_{n_k},u_{n_k}) \geq J(q^\star,u^\star)\,.$$

In particular we have

$$J^\star = \lim_{k\to\infty} J(q_{n_k},u_{n_k}) \geq J(q^\star,u^\star) \text{ and}$$
$$J^\star \leq J(q^\star,u^\star) \text{ from definition of } J^\star\,.$$

□

The proof exposes the inherent traits of optimization with PDEs in infinite-dimensional function spaces, which carry over to shape optimization. In Section B.2.3 a short discussion on the difference to finite-dimensional optimization can be found.

Remark 2.23. Often, the objective function J has the form

$$J(q,u) = \int_\Omega j_1(q,u)\,\mathrm{d}x + \int_\Gamma j_2(q,u)\,\mathrm{d}s + \frac{\alpha}{2}\|q\|_{\mathcal{Q}}^2$$

where j_1 and j_2 are integrable functions. The third summand is a *regularization* or *cost of control* term with whom compactness assumptions on $\mathcal{Q}_{\mathrm{ad}}$ in Theorem 2.22, and thus for well-posedness of the problem, can be shown.

Next, a first order condition for optimal pairs is established.

2.4.3 Optimality System

Our aim is to characterize solutions of optimal control problems of type (2.44) by a necessary first order optimality condition which will be given by the Karush-Kuhn-Tucker (KKT) system. As motivation we look at a finite-dimensional optimization problem.

Example 2.3. Consider the minimization problem

$$\min_{x\in\mathbb{R}^N} \phi(x,g) \text{ subject to } F(x,g(x)) = 0$$

with smooth enough functions ϕ and $F\colon \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$. We try to find a function g that satisfies the implicit function $F(x,g(x)) = 0$ and minimizes ϕ. At a minimum we have a stationary point $(x^\star, y^\star)$ of ϕ with $y^\star = g(x^\star)$ and the chain rule

$$0 \overset{!}{=} \mathrm{d}_x\phi(x^\star,y^\star) = \partial_x\phi(x^\star,y^\star) + \partial_y\phi(x^\star,y^\star)\,\partial_x g(x^\star)$$

holds. Furthermore, by the implicit function theorem the derivative of g follows as

$$\partial_x g(x^\star) = -\big(\partial_y F(x^\star,y^\star)\big)^{-1}\,\partial_x F(x^\star,y^\star)$$

and thus it is the solution to a linear system.

A similar derivation is given for optimal control problems where the PDE constraint $\mathrm{e}(q,u) = 0$ also is an implicit function. The strategy of this section is to

1. first state the chain rule for $J(q,u)$ where u depends on the control, corresponding to $\mathrm{d}_x\phi(x^\star,y^\star)$ in the above finite-dimensional example. For that we have to differentiate a solution operator $\mathrm{S}(q)$ analogously to $\partial_x g(x^\star)$. Eventually, we obtain sensitivities, or directional derivatives, of the state equation and finally of the cost functional.
2. This requires, just as in the finite-dimensional example, solving a linear equation for each control variation. For the PDE and the finite-dimensional example as well, there is a more efficient way which incorporates that we ultimately aim not for $\partial_x g$ but for the product $\partial_y\phi(x^\star,y^\star)\,\partial_x g(x^\star)$.
3. This leads to the adjoint approach which is equivalent to pursuing an optimality system by a Lagrange function. That means, the objective function and constraint are coupled in a Lagrange function where the constraint is in this case the state equation.
4. An optimal solution of (2.44) is then a stationary point of this Lagrangian, i.e. its gradient vanishes at an optimum. This gradient condition describes the Karush-Kuhn-Tucker optimality system, in which also the state equation and the adjoint equation are satisfied.

We begin with the first step from the list above.

Sensitivities

Our goal in this part is to differentiate the cost functional J. This shows us how susceptible the state u and costs J are towards a change in the control q. The derivatives of u and J are therefore also called *sensitivities*. We follow the outline of the finite-dimensional problem; before doing so we state the notation for differentiation in Banach spaces.

Differentiation in Banach spaces In a functional space we need a generalized concept of differentiability, we summarize here from Appendix B.3. Let $j\colon \mathcal{Q} \to \mathbb{R}$ be a real-valued functional defined over a Banach space $\mathcal{Q}$.

- If it exists, we denote its *directional derivative* at q in direction of δq by $\mathrm{d}j(q;\delta q) = \lim_{t\to 0^+} \frac{1}{t}(j(q+t\delta q) - j(q)) = \mathrm{d}_t J(u+th)\Big|_{t=0}$
- If it exists, we denote its *Fréchet-derivative* at q by $j'(q)$ and its *Fréchet-differential* in a direction δq by $\mathrm{d}j(q;\delta q) = j'(q)\delta q$.
- We speak of its *gradient* if $\mathcal{Q}$ is a Hilbert space with inner product $(\cdot,\cdot)$ as the element $\nabla j \in \mathcal{Q}$: $(\nabla j, \delta q) = \mathrm{d}j(q;\delta q)$ for all $\delta q \in \mathcal{Q}$.
- For functionals $\mathrm{e}\colon \mathcal{Q}\times\mathcal{U}$ over two function spaces, e'_q denotes the *partial derivatives* and $\mathrm{d}_q\mathrm{e}$ the partial differentials, as in Definition B.19.

We now proceed with the to-do list from above.

Assumptions Requiring smooth enough functions ϕ and g in the finite-dimensional example is expressed in the optimal control case with following assumptions.

Assumption 2.24. Let $\mathcal{Q}, \mathcal{U}, \mathcal{Z}$ denote Banach spaces and suppose

(*A1*) $J\colon \mathcal{Q}\times\mathcal{U} \to \mathbb{R}$ and $\mathrm{e}\colon \mathcal{Q}\times\mathcal{U} \to \mathcal{Z}$ are continuously Fréchet-differentiable,

(*A2*) there exists a solution operator which assigns to each $q \in \mathcal{Q}$ a unique state $\mathrm{S}\colon q \mapsto u(q) \in \mathcal{U}$ (with Theorem 2.12, Lax-Milgram Lemma),

(*A3*) $\mathrm{e}'_u(q, \mathrm{S}(q)) \in \mathcal{L}(\mathcal{U}, \mathcal{Z})$ is continuously invertible.

See [Hinze et al., 2009, p. 52] for the following handy implication of Assumption 2.24 and the Implicit Function Theorem B.21.

Corollary 2.25. *From Assumption 2.24 it follows that the solution operator* S *is continuously Fréchet-differentiable.*

If a solution operator exists, we can consider the reduced cost functional.

Definition 2.26. For a solution operator S define the *reduced cost functional* $j\colon \mathcal{Q} \to \mathbb{R}$ as $j(q) = J(q, \mathrm{S}(q))$.

It is well-defined since S exists for all controls by assumption (*A2*). Then the optimal control problem (2.44) is equivalent to

$$\min j(q),\ q \in \mathcal{Q}_{\mathrm{ad}}\,. \tag{2.47}$$

Differentiation of state equations We need to differentiate j w.r.t. q to find candidates for an optimum. To get the derivative of the state $u = \mathrm{S}(q)$ we apply the chain rule B.20 for Fréchet-derivatives to the state equation which is allowed because of (*A1*) in Assumption 2.24 and Corollary 2.25

$$0 = \mathrm{e}'(q, \mathrm{S}(q)) = \mathrm{e}'_q(q, \mathrm{S}(q)) + \mathrm{e}'_u(q, \mathrm{S}(q))\,\mathrm{S}'(q) \in \mathcal{L}(\mathcal{Q}, \mathcal{Z})\,. \tag{2.48}$$

For a direction $\delta q \in \mathcal{Q}$, denote the Fréchet-differential $\mathrm{dS}(q;\delta q) = \mathrm{S}'(q)\delta q =: \delta_q u$. Then the differential $\delta_q u$ of the state u is the solution of

$$\mathrm{e}'_u(q,u)\delta_q u = -\,\mathrm{e}'_q(q,u)(\delta q)\,. \tag{2.49}$$

Differentiation of reduced cost functionals Having obtained the derivative of the solution operator, the derivative of the reduced cost functional can be computed as $j'(q) \in \mathcal{L}(\mathcal{Q}, \mathbb{R}) = \mathcal{Q}^*$. Applying the chain rule to J we get that

$$j'(q) = J'_q(q, \mathrm{S}(q)) + J'_u(q, \mathrm{S}(q))\, \mathrm{S}'(q)\,. \tag{2.50}$$

Therefore, the Fréchet-differential of j at q in a direction $\delta q \in \mathcal{Q}$ is given by

$$\mathrm{d}j(q;\delta q) = \langle j'(q), \delta q\rangle_{\mathcal{Q}^*,\mathcal{Q}} = \tag{2.51}$$

$$\overset{(2.50)}{=} \langle J'_u(q, \mathrm{S}(q)), \mathrm{S}'(q)\delta q\rangle_{\mathcal{U}^*,\mathcal{U}} + \langle J'_q(q, \mathrm{S}(q)), \delta q\rangle_{\mathcal{Q}^*,\mathcal{Q}} = \tag{2.52}$$

$$\overset{(2.49)}{=} \langle J'_u(q, u), \delta_q u\rangle_{\mathcal{U}^*,\mathcal{U}} + \langle J'_q(q, u), \delta q\rangle_{\mathcal{Q}^*,\mathcal{Q}}\,. \tag{2.53}$$

We move on to step two and three of the list on page 22 to obtain the optimality system.

Adjoint approach

For each direction $\delta q \in \mathcal{Q}$, the derivative of the reduced cost functional $\mathrm{d}j(q;\delta q)$ requires solving (2.49) to get the sensitivity $\delta_q u$ of the state for (2.53). However, the term of interest is not really $\delta_q u$ but $\langle J'_u(q,u), \delta_q u\rangle_{\mathcal{U}^*,\mathcal{U}}$ which can be written as

$$\begin{aligned}\langle J'_u(q,u), \delta_q u\rangle_{\mathcal{U}^*,\mathcal{U}} &= \langle J'_u(q, \mathrm{S}(q)), \mathrm{S}'(q)\delta q\rangle_{\mathcal{U}^*,\mathcal{U}} = \\ &= \langle (\mathrm{S}'(q))^* J'_u(q, \mathrm{S}(q)), \delta q\rangle_{\mathcal{Q}^*,\mathcal{Q}}\,,\end{aligned} \tag{2.54}$$

introducing the adjoint solution operator $\mathrm{S}^*\colon \mathcal{Q}^* \to \mathcal{U}^*$. Now we use $(A3)$ in Assumption 2.24 together with (2.48) to deduce

$$-\,\mathrm{e}'_u(q, \mathrm{S}(q))^{-1}\, \mathrm{e}'_q(q, \mathrm{S}(q)) = \mathrm{S}'(q)\,. \tag{2.55}$$

Hence, we have for the desired term in (2.54)

$$(\mathrm{S}'(q))^* J'_u(q, \mathrm{S}(q)) = -\,\mathrm{e}'_q(q, \mathrm{S}(q))^* \underbrace{\mathrm{e}'_u(q, \mathrm{S}(q))^{-*} J'_u(q, \mathrm{S}(q))}_{-z}\,. \tag{2.56}$$

The function $z \in \mathcal{Z}^*$ called *adjoint state* and is the solution of the adjoint equation

$$\mathrm{e}'_u(q, \mathrm{S}(q))^* z = -J'_u(q, \mathrm{S}(q))\,. \tag{2.57}$$

We solve this equation once, and then need to only evaluate $\langle \mathrm{e}'_q(q, \mathrm{S}(q))^* z, \delta q\rangle_{\mathcal{Q}^*,\mathcal{Q}}$.

We are now ready to perform the last step of our list and formulate the Karush-Kuhn-Tucker optimality system, which identifies candidates of optimal solutions for the optimal control problem (2.44).

Karush-Kuhn-Tucker optimality system

In the final to-do item of the list on page 22 we derive an optimality system for problem (2.44). Up to now the state equation $\mathrm{e}(q,u) = 0$ has represented a strong formulation of a PDE. However, for numerics later we want to use its variational form instead. Therefore, we state the Karush-Kuhn-Tucker system in both strong and weak formulation and finally tailor it towards our needs.

We use the Lagrange approach where the cost functional and state equation are coupled in the Lagrange functional:

Definition 2.27. Let the *Lagrange functional* for (2.44) be given by

$$\begin{aligned} &\mathcal{L}\colon \mathcal{Q}\times\mathcal{U}\times\mathcal{Z}^* \to \mathbb{R} \\ &\mathcal{L}(q,u,z) = J(q,u) + \langle z, \mathrm{e}(q,u)\rangle_{\mathcal{Z}^*,\mathcal{Z}}\,. \end{aligned} \tag{2.58}$$

Since we have constraints on the control we might have a strict inclusion $\mathcal{Q}_{\mathrm{ad}} \subsetneq \mathcal{Q}$ and we adjust Assumption 2.24 therefor.

Assumption 2.28. $\mathcal{Q}$, $\mathcal{U}$, $\mathcal{Z}$ Banach spaces and $\mathcal{Q}_{\mathrm{ad}} \subset \mathcal{Q}$ nonempty, convex, and closed.

(*A1*) $J\colon \mathcal{Q}\times\mathcal{U}\to\mathbb{R}$, $\mathrm{e}\colon \mathcal{Q}\times\mathcal{U}\to\mathcal{Z}$ continuously Fréchet-differentiable

(*A2*) $\forall q \in V$, $V \subset \mathcal{Q}$ neighborhood of $\mathcal{Q}_{\mathrm{ad}}$, the state equation has a unique solution $u = u(q) \in \mathcal{U}$

(*A3*) $\mathrm{e}'_u(q, \mathrm{S}(q)) \in \mathcal{L}(\mathcal{Q},\mathcal{Z})$ has a bounded inverse for all $q \in V \supset \mathcal{Q}_{\mathrm{ad}}$.

Karush-Kuhn-Tucker system in strong form First order optimality conditions for control-constrained problems of type (2.44) are given by the Karush-Kuhn-Tucker system which we reference from [Hinze et al., 2009, Corollary 1.3, p. 73].

Theorem 2.29.
Let $(q^\star, u^\star)$ be an optimal solution of (2.44) and Assumption 2.28 holds. Then there exists an adjoint state $z^\star \in \mathcal{Z}^*$ such that the following optimality conditions hold.

- *State equation:* $\mathcal{L}'_z(q^\star,u^\star,z^\star) = 0$, *i.e.* $\mathrm{e}(q,u) = 0$
- *Adjoint equation* (2.57)*:* $\mathcal{L}'_u(q^\star,u^\star,z^\star) = 0$, *i.e.* $\mathrm{e}'_u(q^\star,u^\star)^* z^\star = -J'_u(q^\star,u^\star)$
- *Optimality condition:* $q^\star \in \mathcal{Q}_{ad}$, $\langle \mathcal{L}'_q(q^\star,u^\star,z^\star), q-q^\star\rangle_{\mathcal{Q}^*,\mathcal{Q}} \geq 0\ \forall q \in \mathcal{Q}_{ad}$, *or* $\mathcal{L}'_q(q^\star,u^\star,z^\star) = 0$ *if* $\mathcal{Q}_{ad} = \mathcal{Q}$

Remark 2.30. The adjoint $z^\star$ can also be interpreted as Lagrange multiplier, however, we do not pursue this line of thinking here. More information can be found for instance in [Hinze et al., 2009] and [Ito and Kunisch, 2008].

Karush-Kuhn-Tucker system in variational form The KKT system in Theorem 2.29 can be equivalently written in a variational form with dual pairings:

- State equation: $\langle \mathrm{e}(q^\star,u^\star), \delta z\rangle_{\mathcal{Z},\mathcal{Z}^*} = 0\ \forall \delta z \in \mathcal{Z}^*$.
- Adjoint equation: $\langle \mathrm{e}'_u(q^\star,u^\star)^* z^\star, \delta u\rangle_{\mathcal{U}^*,\mathcal{U}} = -\langle J'_u(q^\star,u^\star), \delta u\rangle_{\mathcal{U}^*,\mathcal{U}}\ \forall \delta u \in \mathcal{U}$.
- Optimality condition: $q^\star \in \mathcal{Q}_{\mathrm{ad}}$, $\langle \mathcal{L}'_q(q^\star,u^\star,z^\star), \delta q - q^\star\rangle_{\mathcal{Q}^*,\mathcal{Q}} \geq 0\ \forall \delta q \in \mathcal{Q}$.

In our cases we consider states u described by linear elliptic second order PDEs in their weak form for which we give the KKT system next.

KKT system for our model problems We assume that u is a weak solution of $\mathrm{e}(q,u)$, an elliptic, 2nd order PDE defined over the Hilbert space $\mathcal{U}=\mathcal{V}$. Since Hilbert spaces can be identified with their dual, Theorem B.3, we have $\mathcal{V}=\mathcal{V}^*=\mathcal{Z}=\mathcal{Z}^*$. To note the dependence on the control, we introduce operators A and F corresponding to the bilinear and linear form. I.e., we have $\mathrm{e}(q,u)=\mathrm{F}(q)-\mathrm{A}(q,u)$ where we say that an operator $\mathrm{A}\colon \mathcal{Q}\times\mathcal{V}\to\mathcal{V}^*$ yields for each control a bilinear form $\langle \mathrm{A}(q,u),v\rangle_{\mathcal{V}^*,\mathcal{V}}$ according to the operator notation (2.27). The existence of such operators is ensured by Lemma B.5. For instance the example PDE in (2.42) yields for all $v\in\mathcal{V}=H_0^1(\Omega)$:

$$\langle \mathrm{A}(q,u),v\rangle_{\mathcal{V}^*,\mathcal{V}}=\int_\Omega \nabla u\cdot\nabla v-\nabla q\cdot\nabla uv\,\mathrm{d}x=\int_\Omega fv\,\mathrm{d}x=l(v)=\langle \mathrm{F}(q),v\rangle_{\mathcal{V}^*,\mathcal{V}}\,.$$

The Lagrange functional is with Definition 2.27 given by the sum of cost functional J from (2.43) and the PDE in our example as

$$\mathcal{L}(q,u,z)=\int_\Omega f(x)u(x)\,\mathrm{d}x+\int_\Omega fz\,\mathrm{d}x-\int_\Omega \nabla u\cdot\nabla z-\nabla q\cdot\nabla uz\,\mathrm{d}x$$

or in general by

$$\mathcal{L}(q,u,z)=J(q,u)+\langle \mathrm{F}(q),z\rangle_{\mathcal{V}^*,\mathcal{V}}-\langle \mathrm{A}(q,u),z\rangle_{\mathcal{V}^*,\mathcal{V}}\,. \tag{2.59}$$

Then the sensitivity calculation

- state $\mathrm{d}_z\mathcal{L}(q,u,z;\delta z)=\int_\Omega f\delta z\,\mathrm{d}x-\int_\Omega \nabla u\cdot\nabla\delta z-\nabla q\cdot\nabla u\delta z\,\mathrm{d}x$,
- adjoint $\mathrm{d}_u\mathcal{L}(q,u,z;\delta u)=\int_\Omega f(x)\delta u(x)\,\mathrm{d}x-\int_\Omega \nabla\delta u\cdot\nabla z-\nabla q\cdot\nabla\delta uz\,\mathrm{d}x$,
- optimality condition $\mathrm{d}_q\mathcal{L}(q,u,z;\delta q)=\int_\Omega \nabla\delta q\cdot\nabla uz\,\mathrm{d}x$.

and Lagrangian approach yield the KKT system for Example 2.2. In the general case for linear elliptic second order PDEs the KKT system in the variational form is given next by a corollary of Theorem 2.29.

Corollary 2.31. *Let $(q^\star,u^\star)$ be an optimal solution of (2.44) and Assumption 2.28 holds. Then there exists an adjoint state $z^\star\in\mathcal{V}$ such that the following optimality system holds.*

- *The state equation holds. For $u^\star\in\mathcal{V}$:*

$$\langle \mathrm{F}(q^\star),\delta z\rangle_{\mathcal{V}^*,\mathcal{V}}=\langle \mathrm{A}(q^\star,u^\star),\delta z\rangle_{\mathcal{V}^*,\mathcal{V}}\quad\forall\delta z\in\mathcal{V}\,.$$

- *The adjoint equation holds. For $z^\star\in\mathcal{V}$:*

$$\langle \mathrm{F}'_u(q^\star)(z^\star)-\mathrm{A}'_u(q^\star,u^\star)(z^\star),\delta u\rangle_{\mathcal{V}^*,\mathcal{V}}=-\langle J'_u(q^\star,u^\star),\delta u\rangle_{\mathcal{V}^*,\mathcal{V}}\quad\forall\delta u\in\mathcal{V}\,.$$

- *The first order necessary optimality condition holds. For $q^\star\in\mathcal{Q}_{ad}$:*

$$\langle \mathcal{L}'_q(q^\star,u^\star,z^\star),\delta q-q^\star\rangle_{\mathcal{Q}^*,\mathcal{Q}}\geq 0\quad\forall\delta q\in\mathcal{Q}\,.$$

The last corollary provides the basis for the transformation approach in shape optimization later. In particular the last item, the optimality condition, is the way to obtain a shape derivative: In the Lagrange functional the variables are independent. Therefore, they are independent in the cost term J appearing in $\mathcal{L}$ and we do not consider that the state u depends on the control q, the chain rule is taken care of with the adjoint z. In the end, we use that

$$\langle \mathcal{L}'_q(q^\star, u^\star, z^\star), \delta q - q^\star \rangle_{\mathcal{Q}^*,\mathcal{Q}} = \langle J'_q(q^\star, u^\star), \delta q - q^\star \rangle_{\mathcal{Q}^*,\mathcal{Q}} \quad \forall \delta q \in \mathcal{Q}\,.$$

Remark 2.32. Homogeneous Dirichlet boundary conditions are a bit delicate in this approach because when using $\mathcal{V} = H_0^1(\Omega)$ its dual is $\mathcal{V}^* = H^{-1}(\Omega)$.

Impact on shape optimization We flash forward to the next chapter and shape optimization: Basically, we translate shape optimization problems with perturbation of identity to the optimal control problem (2.44). Since without further ado a set of domains $\mathcal{O}$ forms no Banach space for which we have the results above, one considers instead a space of diffeomorphisms $\tau = id + q$ with q from a Banach space $\mathcal{Q}$, whose images yield the very set of domains $\mathcal{O} = \{\tau(\Omega) = (id + q)(\Omega),\, q \in \mathcal{Q}\}$. Eventually, the control space $\mathcal{Q}$ is identified with the space of transformations and the operators $\mathrm{A}(q, u)$ and $\mathrm{F}(q)$ are the transformed bilinear and linear forms $\hat{a}(\tau)(\hat{u}, \hat{v})$ and $\hat{l}(\tau)(\hat{v})$ in (2.35)–(2.37) from Section 2.3.3 due to a change of variables.

The Lagrangian approach which leads to the KKT system plays a significant role because in this system and especially in the partial derivative $\mathcal{L}'_q$ in Theorem 2.29 and following, the derivative $\delta_q u$ of the state u does not occur anymore. However, usually the differentiability of the state is assumed. In contrast to that, [Ito and Kunisch, 2008] showed that the approach is also applicable to optimization problems where the state variable cannot be differentiated w.r.t. the control. Especially for shape optimization this means that even though the state is not differentiable with respect to the shape we can obtain the shape derivative of the cost functional, i.e. the KKT system.

3
Abstract Shape Optimization Framework

In this chapter we study the shape optimization problems (1.1) and how to derive first-order optimality conditions for them. This yields in particular a shape gradient that can be used as descent direction in a numerical optimization of our applications in Chapter 6. As one of the basic references in shape optimization we refer to [Murat and Simon, 1976a], [Pironneau, 1983], [Sokolowski and Zolésio, 1992] and [Haslinger and Mäkinen, 2003]. Additionally, we point to [Allaire and Jouve, 2005] and [Bendsøe and Sigmund, 2003] for the homogenization approach and topology optimization.

Since shape optimization is a special case in optimal control, it leans on Chapter 2.4. As usual in optimal control, SOPs are posed in infinite-dimensional function spaces. Likewise, we are compelled to decide for the numerical realization later if we derive optimality conditions before or after the discretization. We compare both variants in Chapter 4 for IGA, however, the foundation for optimizing first is given in this chapter.

The analysis of the infinite-dimensional problem matters because it can give more information than a discrete version: In [Haslinger and Mäkinen, 2003, p. 11 ff.] and also in [Allaire and Henrot, 2001], as an example, a finite-dimensional problem may have a solution when the original (i.e. the infinite) has none.

Our targeted optimality system and shape gradient do not carry information about numerical discretization like meshes and geometry parameterization yet. However, in view of isogeometric shape optimization later we propose a formulation which already considers that in IGA all domains are parameterizations over a fixed parameter domain. These transformations are then discretized by B-splines or NURBS in Chapter 4. We discuss our abstract frame in the light of similar transformation approaches such as [Eppler et al., 2007], [Ito et al., 2008], [Brandenburg et al., 2009], and [Kiniger, 2015].

The chapter is organized as follows. In **Section 3.1** we first introduce shape optimization problems over a set of domains $\mathcal{O}$ to fix the notation and highlight the challenges arising for these problems. Subsequently, we use the classical perturbation of identity approach from [Murat and Simon, 1976b] to construct a metric on $\mathcal{O}$ in **Section 3.2** and provide a differential structure with which first specimen of shape derivatives are shown. This is as much a guideline as a basis to our framework: We use this method in **Section 3.3** to reformulate the general shape optimization problem over a fixed parameter domain. This point of view absorbs isogeometric analysis and results in a scheme to obtain IGA-typical shape derivatives. Finally, we can give the optimality system for the SOP in its optimal control form in an isogeometric analysis suitable way **Section 3.4**.

3.1 Shape Optimization Problems

In this section we add some amendments like necessary function spaces to the formulation of shape optimization problems (1.1) in the introduction to this thesis to make them well-defined. Moreover, we briefly touch on the question of existence of optimal shapes.

Problem formulation

From the manifold of applications also arise different classes of shape optimization problems—only a fraction of which is treated in this work. First of all, we exclude *topology optimization*. That is, we do not allow the generation of new holes during the optimization process, for a comparison see Figure 3.1. Topology optimization requires different techniques and does not indicate to benefit from tensor-product structured discretizations inherent to IGA.

(a) Initial domain **(b)** Geometric optimization **(c)** Topology optimization

Figure 3.1: Difference of geometric and topological optimization on an initial design: Topological optimization allows new holes.

We recall the shape optimization problem (1.1) from the introduction

$$\min J(\Omega, u) \quad \text{s.t.} \quad \mathrm{e}(\Omega, u) = 0 \quad \text{and } \Omega \in \mathcal{O}_{\mathrm{ad}}\,, \tag{3.1}$$

for admissible domains $\Omega \in \mathcal{O}_{\mathrm{ad}} \subset \mathscr{P}(\mathbb{R}^d)$, $d = 2, 3$, under a constraint in form of a second order linear elliptic PDE $\mathrm{e}(\Omega, u) = 0$ from Section 2.2. We say that $\mathscr{P}(S)$ is the *power set* of a set S, i.e. it contains all subsets of S. Such problems cannot be analyzed immediately because the usual tools like continuity and differentiation are, at first, not available on $\mathscr{P}(\mathbb{R}^d)$. Here, we fix the notation of shape optimization problems to tackle shape continuity and differentiation in the next section.

To say that we solve a PDE over various domains, we state the PDE operator $\mathrm{e}(\cdot, \cdot)$ more precisely.

Definition 3.1. Let $\mathcal{O}_{\mathrm{ad}}$ denote a set of bounded domains in $\mathbb{R}^d$. For any $\Omega \in \mathcal{O}_{\mathrm{ad}}$ let $\mathcal{U}(\Omega)$ and $\mathcal{Z}(\Omega)$ define Banach spaces of functions over Ω. Then

$$\mathrm{e}\colon \{(\Omega, u)\colon \Omega \in \mathcal{O}_{\mathrm{ad}},\ u \in \mathcal{U}(\Omega)\} \to \{z \in \mathcal{Z}(\Omega), \Omega \in \mathcal{O}_{\mathrm{ad}}\} \tag{3.2}$$

is an operator between them where $\mathrm{e}(\Omega, u) = 0$ solves a PDE posed over Ω as in Section 2.2. The spaces $\mathcal{U}(\Omega)$ and $\mathcal{Z}(\Omega)$ must be chosen carefully for it to be well-defined. For fixed domains, the PDE operator notation is explained on page 13.

Consistent data In order to make PDE data in $\mathrm{e}(\cdot,\cdot)$ like $\mathrm{L}u - f = 0$ well defined for all Ω we assume that there is a *holding-all* $D \subset \mathbb{R}^d$ for which $\Omega \subset D$ for all $\Omega \in \mathcal{O}$. For instance let then f be defined over D and in $\mathrm{L}u - f = 0$ we actually consider f restricted to Ω. We can now better specify an SOP.

Definition 3.2. For a holding-all $D \subset \mathbb{R}^d$, a bounded domain, let $\mathcal{O}_{\mathrm{ad}} \subset \mathcal{O} \subset \mathscr{P}(D)$ be a set of bounded, admissible domains $\Omega \subset D$, and denote by $\mathcal{U}(\Omega)$ and $\mathcal{Z}(\Omega)$ Banach spaces over $\Omega \in \mathcal{O}_{\mathrm{ad}}$. In $\mathcal{O}_{\mathrm{ad}}$ additional constraints can be formulated. The functional J is the real valued objective function defined over

$$J\colon \{(\Omega, u)\colon \Omega \in \mathcal{O}_{\mathrm{ad}},\ u \in \mathcal{U}(\Omega)\} \to \mathbb{R}\,,$$

and the operator $\mathrm{e}(\cdot,\cdot)$ is given by Definition 3.1. Then an abstract SOP (3.1) over a set of domains takes the form

$$\begin{cases} \text{find } \Omega^\star \in \mathcal{O}_{\mathrm{ad}} \text{ such that} \\ J(\Omega^\star, u(\Omega^\star)) \le J(\Omega, u(\Omega)) \text{ and } \mathrm{e}(\Omega, u) = 0\,. \end{cases} \tag{$\mathcal{P}$}$$

The notation in Definition 3.1 stems from [Brandenburg, 2011] and is motivated by optimal control problems (2.44). There, $\mathcal{O}$ is replaced by a control space $\mathcal{Q}$ and $q \in \mathcal{Q}$ is for instance a parameter in the PDE. A comparison between such standard optimal control problems and SOPs $(\mathcal{P})$ immediately reveals the crux in shape optimization: In (2.44) $\mathrm{e}(\cdot,\cdot)$ is defined over $\mathcal{Q}_{\mathrm{ad}} \times \mathcal{U}$ to $\mathcal{Z}$, where typically Ω is fixed and thus are the Banach spaces $\mathcal{U}$ and $\mathcal{Z}$, unlike in (3.2), where $\{\mathcal{U}(\Omega),\ \Omega \in \mathcal{O}\}$ and $\{\mathcal{Z}(\Omega),\ \Omega \in \mathcal{O}\}$ are families of function spaces.

The existence of optimal shapes in the problem formulation over a set of domains is studied next.

Existence of optimal shapes

Suppose $\mathcal{O}$ is a complete, metric space, i.e. for a sequence of domains $\{\Omega_n\} \subset \mathcal{O}$ the convergence $\Omega_n \to \Omega$ to an element $\Omega \in \mathcal{O}$ makes sense.

Assumption 3.3. We assume that

- $(A1)$ $\mathcal{O}_{\mathrm{ad}} \subset \mathcal{O}$ is compact.
- $(A2)$ The problem $(\mathcal{P})$ has a feasible point.
- $(A3)$ To each $\Omega \in \mathcal{O}$ exists a unique solution $u(\Omega) \in \mathcal{U}(\Omega)$ of the state equation.
- $(A4)$ The solutions $u(\Omega)$ of the state equation depend continuously on $\Omega \in \mathcal{O}_{\mathrm{ad}}$, i.e. $\Omega_n \to \Omega \Rightarrow u_n \to u$ with some sort of convergence for $u_n \in \mathcal{U}(\Omega_n)$ across different function spaces to $u \in \mathcal{U}(\Omega)$.
- $(A5)$ J is sequentially weakly lower semicontinuous.

Denote the *feasible set* of $(\mathcal{P})$ as

$$\mathcal{W}_{\mathrm{ad}} := \{(\Omega, u)\colon \Omega \in \mathcal{O}_{\mathrm{ad}},\ u = u(\Omega) \text{ solves the state equation}\}\,. \tag{3.3}$$

Then [Haslinger and Mäkinen, 2003] boils the existence of optimal domains for $(\mathcal{P})$ down to fulfilling the above assumptions.

Theorem 3.4.
Under Assumption 3.3 the shape optimization problem $(\mathcal{P})$ has at least one solution.

Proof. We state and comment on [Haslinger and Mäkinen, 2003, Theorem 2.10, p. 47]. From $(A2)$ we know $\mathcal{W}_{\text{ad}} \neq \emptyset$ and

$$J^\star := \inf_{(\Omega,u)\in\mathcal{W}_{\text{ad}}} J(\Omega, u) \text{ exists.}$$

There is a minimizing sequence $(\Omega_n, u_n) \subset \mathcal{W}_{\text{ad}}$ such that $\lim_{n\to\infty} J(\Omega_n, u_n) = J^\star$. Since $\mathcal{O}_{\text{ad}}$ is compact, $(A1)$, we can extract a subsequence $\{\Omega_{n_k}\}$ and $\Omega^\star \in \mathcal{O}_{\text{ad}}$ such that

$$\Omega_{n_k} \to \Omega^\star, \quad k \to \infty\,.$$

Therefore, $(A4)$ ensures that also $u(\Omega_{n_k}) \to u(\Omega^\star), \quad k \to \infty$, and because of $(A5)$

$$\liminf_{k\to\infty} J(\Omega_{n_k}, u(\Omega_{n_k})) \geq J(\Omega^\star, u(\Omega^\star))$$

which concludes the proof by definition of $J^\star$. □

Note, that we structured the assumptions to match those for the standard optimal control case in Theorem 2.22 where similar concepts are used. The existence result for SOPs from [Haslinger and Mäkinen, 2003] is kept very general, however, a metric on $\mathcal{O}$ is established with perturbation of identity next. Checking $(A1)$ and $(A4)$ in Assumption 3.3 is not trivial; the continuity across function space families is realized with extending u_n from Ω_n to the holding all D and then convergence in the Lebesgue-norm via characteristic functions is shown. For the compactness of the shape "space" [Haslinger and Mäkinen, 2003] use functions for the boundary to characterize a shape. Under uniform convergence of these, the Arzelà-Ascoli theorem yields compactness.

The next section provides a vector space structure for the set of shapes in which domains are "points", and a differential structure to calculate shape sensitivities.

3.2 Perturbation of Identity Method

We follow [Murat and Simon, 1976b] or their summary [Murat and Simon, 1976a] to introduce the method of perturbation of identity. A particularly helpful overview of the method is given in [Brandenburg, 2011, Lindemann, 2012].

To set the course of this section, we pretend for a moment that Ω is a point, i.e. $\Omega \in \mathbb{R}^d$, and consider the usual definition of a functional $J\colon \mathcal{O} \to \mathbb{R}$ continuous at Ω and its linearization with its derivative for $\|\delta\Omega\| \to 0$

$$\lim_{\Omega'\to\Omega} J(\Omega) = J(\Omega) \text{ and } J(\Omega + \delta\Omega) = J(\Omega) + J'(\Omega)\delta\Omega + o(\|\delta\Omega\|)\,.$$

So for shape continuity of J we need a notion of convergence $\Omega' \to \Omega$ which is given in case there is a metric d on $\mathcal{O}$

$$\Omega' \to \Omega \Leftrightarrow \mathrm{d}(\Omega', \Omega) \to 0 \text{ in } \mathbb{R}\,.$$

Moreover, the derivative necessitates a linear space structure with an addition $\Omega + \delta\Omega$ and a norm on $\mathcal{O}$. All shape optimization methods somehow have to provide these structures; we use the well-known perturbation ansatz from [Murat and Simon, 1976a].

To find a metric, a suitable space of transformations $\mathscr{C}$ is used to express the shapes $\mathcal{O} \subset \{\Omega = \tau(\Omega_0) \colon \tau \in \mathscr{C}\}$ and the metric from this space induces the one on $\mathcal{O}$. To take control over domain variations, the transformations are used to perturb the current domain. The differential structure can then be induced on $\mathcal{O}$ through the perturbations. Thus, by considering variations of functions, the SOP is shifted to an optimal control setting and concepts from Section 2.4 can be used to assert existence results and more importantly to derive optimality conditions.

In this section we proceed as follows. First we introduce a metric on the set of domains by considering domains resulting from transformations $\tau \in \mathscr{C}$ of a reference domain. This allows to define a sort of shape continuity of functions $\phi(\Omega_n) \to \phi(\Omega)$ if $\Omega_n \to \Omega$ for $n \to \infty$ by considering $\tau_n \to \tau$. Next, a notion of differentiability is established, i.e. instead of $\partial_\Omega J$ we use $\partial_\tau J$.

3.2.1 Metric structure

For a positive integer k define the space of transformation of regularity k

$$\mathscr{V}^k := \left\{\tau \colon \mathbb{R}^d \to \mathbb{R}^d \text{ with } \tau = id + q, \text{ where } q \in W^{k,\infty}(\mathbb{R}^d)^d\right\} \tag{3.4}$$

and denote a set of *perturbations of identity* by

$$\mathscr{C}^k := \left\{\tau \colon \mathbb{R}^d \to \mathbb{R}^d \text{ bijection, } \tau,\, \tau^{-1} \in \mathscr{V}^k\right\}. \tag{3.5}$$

Then let $\Omega_0 \subset \mathbb{R}^d$ be an open, connected reference domain and denote by

$$\mathscr{D}^k_{\Omega_0} := \{\Omega \subset \mathbb{R}^d \colon \Omega = \tau(\Omega_0) \text{ with } \tau \in \mathscr{C}^k\} \tag{3.6}$$

the set of all domains resulting from transformations under $\mathscr{C}^k$. For a fixed $k > 0$ and norm $\|.\| = \|.\|_{W^{k,\infty}(\mathbb{R}^d)^d}$ the map $\tilde{\mathrm{d}}$ on $\mathscr{D}^k_{\Omega_0} \times \mathscr{D}^k_{\Omega_0}$ defined by

$$\tilde{\mathrm{d}}(\Omega_1, \Omega_2) = \inf_{\substack{\tau(\Omega_1)=\Omega_2,\\ \tau \in \mathscr{C}^k}} \{\|\tau - id\| + \|\tau^{-1} - id\|\} \tag{3.7}$$

is a pseudo-distance on $\mathscr{D}^k_{\Omega_0}$ as the triangle inequality does not hold. However, with the following statement from [Murat and Simon, 1976a, Theorem 1, p. 56] this can be mended.

Theorem 3.5.
For fixed $k > 0$ there exists a constant $\eta > 0$ such that

$$\mathrm{d}(\Omega_1, \Omega_2) := \inf\{\sqrt{\tilde{\mathrm{d}}}, \eta\} \tag{3.8}$$

defines a metric on $\mathscr{D}^k_{\Omega_0}$ and $(\mathscr{D}^k_{\Omega_0}, \mathrm{d})$ is a complete, metric space.

A perturbation $\tau \in \mathscr{C}^k$ has the form $\tau = id + q$ where $q \in W^{k,\infty}(\mathbb{R}^d)^d$. So basically, the desired distance between two shapes in Theorem 3.5 is induced by the norm of the space of domain variations, i.e. by the vector space $W^{k,\infty}(\mathbb{R}^d)^d$ which contains the perturbations q.

The next lemma asserts that $\Omega_n \to \Omega$ if $q_n \to q$.

Lemma 3.6. *Let $\Omega \in \mathscr{D}^k_{\Omega_0}$ and $k > 0$, Ω_0 open in $\mathbb{R}^d$. Then for $r > 0$ small enough, $q \mapsto (id + q)(\Omega)$ transforms a neighborhood of $0 \in W^{k,\infty}(\mathbb{R}^d)^d$ given by the open ball $B_{W^{k,\infty}(\mathbb{R}^d)^d}(0, r)$ to a neighborhood of Ω given by the open ball $B_{\mathscr{D}^k_{\Omega_0}}(\Omega, r) = \{\Omega' \in \mathscr{D}^k_{\Omega_0} : \mathrm{d}(\Omega, \Omega') < r\}$.*

Proof. Proof in [Murat and Simon, 1976b, Lemma 3.1, p. III-2]. □

Remark 3.7. Typically, we do not consider the convergence of Ω_k anymore from now on but of q_k or τ_k instead. Moreover, we note that Ω_0 served to define the shape space $\mathscr{D}^k_{\Omega_0}$ and its metric. In the perturbation of identity method, it has no special position furthermore.

3.2.2 Differential structure

We now define a notion of derivatives w.r.t. domains. As done for the distance function the question is recast in the space of perturbations. Instead of maps defined over a set of domains, like $J : \mathcal{O} \times \{\mathcal{U}(\Omega),\ \Omega \in \mathcal{O}\} \to \mathbb{R}$, consider functionals over the space of perturbations $\mathscr{V}^k$ from (3.5) which yield $\mathcal{O}$. The Sobolev space $W^{k,\infty}(\mathbb{R}^d)^d$ has already the desired linear structure and so has the affine space $\mathscr{V}^k$. We use this to define shape differentiability by extending the notion of Gâteaux- and Fréchet derivatives to sets (domains) as [Murat and Simon, 1976a, Def. 3.1, p. III-3].

Definition 3.8. Let X denote a Banach space and ϕ a map from an open set $\mathcal{O} \subset \mathscr{D}^k_{\Omega_0}$ where $k > 0$ with values in X, $\phi : \mathcal{O} \to X$. We say ϕ is *(shape) differentiable* at $\Omega \in \mathcal{O}$ if in a neighborhood of 0 in $W^{k,\infty}(\mathbb{R}^d)^d$ the map

$$\tilde{\phi} : q \mapsto \phi\big((id + q)(\Omega)\big) \tag{3.9}$$

is defined and is Fréchet-differentiable at 0. Then, the *(shape) derivative* of ϕ is defined by

$$\phi'(\Omega) := \tilde{\phi}'(0) \in \mathcal{L}\left(W^{k,\infty}(\mathbb{R}^d)^d, X\right). \tag{3.10}$$

Note that this means also for the differential at Ω in direction of $\delta q \in W^{k,\infty}(\mathbb{R}^d)^d$

$$\mathrm{d}\phi(\Omega; \delta q) := \mathrm{d}\tilde{\phi}(0; \delta q) = \phi'(\Omega)\delta q\,. \tag{3.11}$$

In Definition 3.8 of shape derivatives it was possible to require Gâteaux-differentiability of ϕ instead of Fréchet, however, then the chain rule for composite functions does not hold anymore, which is important though for deriving necessary optimality conditions, see [Delfour and Zolésio, 2011, p. 458] or [Murat and Simon, 1976a, Remark 3.2]. It is especially vital for isogeometric shape optimization where we always consider compositions with parameterizations G of the current domain Ω. Therefore the chain rule for perturbations of identity is very important for shape optimization with IGA.

Lemma 3.9. *Let $\phi : \mathcal{O} \to X$ be a shape functional with values in a Banach space X where $\mathcal{O} \subset \mathscr{D}^k_{\Omega_0}$ with $k > 0$. Further denote by $F : Y \to \mathcal{O}$ a map from a Banach space Y onto the shape space. If F is differentiable at y and if ϕ is differentiable at $\Omega = F(y)$ in the sense of Definition 3.8 then $\phi \circ F$ is differentiable from Y to X at y and*

$$(\phi \circ F)'(y)\delta y = \phi'(\Omega)\big(F'(y)\delta y\big)\,.$$

Proof. Proof in [Murat and Simon, 1976a, Prop. 3.3, p. III-11]. □

Fréchet-differentiable functions are also directionally differentiable, see also Definition B.16. For practical purposes that is the derivative we evaluate and for which we now introduce the notation.

Directional shape derivatives Let $\phi(\Omega)$ be a given shape functional, then the shape derivative at Ω or at $q = 0$ respectively in direction of δq is defined with perturbation of identity in Definition 3.8 as

$$\mathrm{d}\phi(\Omega; \delta q) := \mathrm{d}\tilde{\phi}(0; \delta q)\,.$$

For directional derivatives at Ω and $q = 0$ respectively, we look at domain variations in a fixed direction $\delta q \in W^{k,\infty}(\mathbb{R}^d)^d$ with parameter $t \geq 0$ and

$$\tau_t := id + t\delta q, \quad \Omega_t := \tau_t(\Omega), \quad \tau_0 = id\,. \tag{3.12}$$

Then we have an equivalent formulation for the directional derivatives at Ω

$$\begin{aligned} \mathrm{d}\phi(\Omega; \delta q) &= \mathrm{d}_t \phi(\tau_t(\Omega))\big|_{t=0} = \lim_{t \downarrow \to 0} \frac{1}{t}\Big(\phi(\tau_t(\Omega)) - \phi(\tau_0(\Omega))\Big) \\ &= \lim_{t \downarrow 0} \frac{1}{t}(\phi(\Omega_t) - \phi(\Omega)) \text{ or} \\ &= \mathrm{d}_t \tilde{\phi}(t\delta q)\big|_{t=0}\,. \end{aligned} \tag{3.13}$$

We apply the definition of shape derivatives to some important shape functionals in the following.

3.2.3 Shape derivatives

In this section we state some shape derivatives of functions of interest obtained with perturbation of identity. Note, that they coincide with those obtained by the speed method in [Sokolowski and Zolésio, 1992]. We also state the result from there for the sake of completeness. Before considering differentiating cost functionals in full, we state some properties of the perturbations of identity.

Lemma 3.10. *Let $\tau \in \mathscr{C}^k$ for $k > 0$. It holds:*

(i) $\phi(\tau) := |\det \mathrm{J}_\tau|$ is differentiable from $\mathscr{V}^k$ to $W^{k-1,\infty}(\mathbb{R}^d)$. Moreover, we have for $\tau = \tau(q) = id + q$ that $\tilde{\phi}(q) := \phi \circ \tau(q)$ is differentiable with the chain rule: For all $\delta q \in W^{k,\infty}(\mathbb{R}^d)^d$

$$\begin{aligned} \mathrm{d}\tilde{\phi}(q; \delta q) &= \mathrm{d}\phi(\tau(q); \mathrm{d}\tau(q; \delta q)) = \\ &= |\det \mathrm{J}_\tau|(\mathrm{div}(\delta q \circ \tau^{-1})) \circ \tau = |\det \mathrm{J}_\tau| \,\mathrm{tr}(\mathrm{J}_{\delta q}\, \mathrm{J}_\tau^{-1})\,. \end{aligned} \tag{3.14}$$

(ii) The map to the inverse Jacobian of the perturbation $\phi(\tau) := \mathrm{J}_\tau^{-1}$ is differentiable from $\mathscr{V}^k$ to $W^{k-1,\infty}(\mathbb{R}^{d,2d})$. With the chain rule we have that for all $\delta q \in W^{k,\infty}(\mathbb{R}^d)^d$ the differential of $\tilde{\phi}(q) = \phi(\tau(q))$ is

$$\mathrm{d}\tilde{\phi}(q; \delta q) = -\,\mathrm{J}_\tau^{-1}\, \mathrm{J}_{\delta q}\, \mathrm{J}_\tau^{-1}\,. \tag{3.15}$$

Proof. The lemma summarizes Lemma 4.2 and 4.3 from [Murat and Simon, 1976b]. □

Note, that $\tau = id + q$ is the identity for $q = 0$. Therefore, with perturbation of identity, the directional derivative in Lemma 3.10

(i) for $\tilde{\phi}(q) = |\det \mathrm{J}_{\tau(q)}|$ is given as $\mathrm{d}\tilde{\phi}(0;\delta q) = \operatorname{div} \delta q$,

(ii) for $\tilde{\phi}(q) = \mathrm{J}^{-1}_{\tau(q)}$ is given as $\mathrm{d}\tilde{\phi}(q;\delta q) = -\mathrm{J}_{\delta q}$.

The determinant of the Jacobian appears from a change of variables in integrals and has to be differentiated to obtain the shape sensitivities of such domain or boundary integrals next.

Shape derivatives of domain integrals

In this section we study shape functionals where the domain of integration is the variable

$$J(\Omega) = \int_{\Omega} \phi(\Omega)(x)\,\mathrm{d}x\,. \tag{3.16}$$

We assume that ϕ is defined over $\mathscr{D}^k_{\Omega_0}$ for $k > 0$ and that it is integrable, $\phi(\Omega) \in L^1(\Omega)$. As in (3.9) consider the transported function $\tilde{\phi}(q) = \phi\big((id+q)(\Omega)\big) \circ (id+q)$ for a fixed Ω which is also in $L^1(\Omega)$ by a change of variables, and especially $\tilde{\phi}(0) = \phi(\Omega)$. Assume that $\tilde{\phi}(q)$ is Fréchet-differentiable from $W^{k,\infty}(\mathbb{R}^d)^d$ to $L^1(\Omega)$. Then we have the shape derivative defined by (3.9) in direction of $\delta q \in W^{k,\infty}(\mathbb{R}^d)^d$ at $q = 0$ or at Ω respectively

$$\mathrm{d}J(\Omega;\delta q) = \int_{\Omega} \phi(\Omega)\operatorname{div}\delta q + \mathrm{d}\tilde{\phi}(0;\delta q)\,\mathrm{d}x\,. \tag{3.17}$$

Proof. For $q \in W^{k,\infty}(\mathbb{R}^d)^d$ in a neighborhood of 0 set $\tau(q) = id + q$ and consider that

$$J\big((id+q)(\Omega)\big) = \int_{\tau(q)(\Omega)} \phi\big(\tau(q)(\Omega)\big)(x)\,\mathrm{d}x\,. \tag{3.18}$$

In a neighborhood of 0 these perturbations of identity $\tau \in \mathscr{C}^k$ behave like diffeomorphisms in the sense that a change of variables holds due to Theorem 2.17

$$\tilde{J}(q) := \int_{\Omega} \phi\big(\tau(q)(\Omega)\big)\big(\tau(q)(x)\big)|\det \mathrm{J}_{\tau(q)}|\,\mathrm{d}x = \int_{\Omega} \tilde{\phi}(q)|\det \mathrm{J}_{\tau(q)}|\,\mathrm{d}x\,. \tag{3.19}$$

Since $\tilde{\phi}$ is differentiable by assumption and $|\det \mathrm{J}_\tau|$ by Lemma 3.10, so is $\tilde{J}$ with

$$\mathrm{d}\tilde{J}(q;\delta q) = \int_{\Omega} \Big(\mathrm{d}\tilde{\phi}(q;\delta q) + \operatorname{div}(\delta q \circ \tau(q)^{-1}) \circ \tau(q)\Big)\,|\det \mathrm{J}_{\tau(q)}|\,\mathrm{d}x \tag{3.20}$$

and at $q = 0$ we have the shape derivative $\mathrm{d}J(\Omega;\delta q)$ which yields (3.17), because then $\tau(0) = id = \tau(0)^{-1}$ and $\det \mathrm{J}_{\tau(0)} = 1$. We refer to [Murat and Simon, 1976b], Theorem 4.1, for the original proof. □

Example 3.1. Assume in (3.17) that the dependence of a function $\phi \in W^{1,1}(\mathbb{R}^d)$ to the shapes is given only by its restriction to Ω,

$$\phi(\Omega) = \phi|_\Omega \,.$$

Then $\tilde{\phi}(q) = \phi \circ (id + q)$ over Ω and it is differentiable at 0 from $W^{k,\infty}(\mathbb{R}^d)^d$ to $L^1(\Omega)$ with $\mathrm{d}\tilde{\phi}(0;\delta q) = \nabla\phi \cdot \delta q$. In particular, this leads to the explicit representation

$$\mathrm{d}J(\Omega;\delta q) = \int_\Omega \phi \operatorname{div} \delta q + \nabla\phi \cdot \delta q \,\mathrm{d}x = \int_\Omega \operatorname{div}(\phi \delta q)\,\mathrm{d}x \tag{3.21}$$

where the latter results from identity (A.1) in the Appendix for the divergence operator.

Remark 3.11. We note here that the chain rule and directional derivative lead to the derivative of $\tilde{\phi}(q)$: $\mathrm{d}\tilde{\phi}(0;\delta q) = \mathrm{d}_t \phi \circ (id + t\delta q)|_{t=0} = \nabla\phi \cdot \delta q$. The more demanding Fréchet-differentiability is proved in [Murat and Simon, 1976b].

Similarly to the above, we next state the shape derivatives of boundary integrals.

Shape derivatives of boundary integrals

In case of shape functionals depending on the integrals over the boundary of a domain

$$J(\Omega) = \int_{\partial\Omega} \phi(\Omega)(s)\,\mathrm{d}s \tag{3.22}$$

where $\Omega \in \mathscr{D}^k_{\Omega_0}$ we state the result from [Murat and Simon, 1976a, Thm. 4.3, p. IV-39]: Let Ω_0 be a open, bounded subset of $\mathbb{R}^d$ with a boundary that is locally Lipschitz and $k > 1$ such that $\phi(\Omega) \in L^1(\partial\Omega)$. If for $\Omega \in \mathcal{O} \subset \mathscr{D}^k_{\Omega_0}$ the functional

$$\tilde{\phi}(q) = \phi((id + q)(\Omega)) \circ (id + q)$$

is differentiable, then we have for all directions $\delta q \in W^{k,\infty}(\mathbb{R}^d)^d$

$$\mathrm{d}J(\Omega;\delta q) = \int_{\partial\Omega} \mathrm{d}\tilde{\phi}(0;\delta q) + \phi(\Omega)(\operatorname{div}\delta q - n^\mathsf{T} \mathrm{J}^\mathsf{T}_{\delta q}\, n)\,\mathrm{d}s\,. \tag{3.23}$$

Shape derivatives of state equation

In both cost functionals (3.16) and (3.22) the integrand ϕ is fairly general and can be in some cases just a restriction to Ω as in Example 3.1. However, in the presence of terms like a state $u = u(\Omega)$ as solution from $\mathrm{e}(\Omega, u) = 0$ the derivative $\mathrm{d}\tilde{\phi}(0;\delta q)$ needs more attention. It requires the derivative of u, i.e. of the control-to-state operator $\mathrm{S}(q) = u(id + q) \circ (id + q)$, which is obtained as for usual optimal control problems by using the implicit function theorem. Therefore, by (2.49) the derivative is given as solution to

$$\mathrm{e}'_u(\Omega, u)\delta_q u = -\,\mathrm{e}'_q(\Omega, u)(\delta q)\,,$$

where $\delta_q u = \mathrm{dS}(q;\delta q) = \mathrm{S}'(q)\delta q$ is the sensitivity of the state w.r.t. changes in the shape. However, to find e'_q and e'_u one has to use the variational forms since the

control occurs in the domains of integrations. We do not state the shape derivatives here because we circumvent them with the Lagrange formalism in the next section. They can be found for the perturbation of identity method in Chapters 5 and 6 of [Murat and Simon, 1976a] for Poisson problems with Neumann or inhomogeneous Dirichlet boundary conditions, though.

Shape gradients

Up to now we have considered only shape derivatives in the direction of some variations like $\mathrm{d}J(\Omega;\delta q)$ in Example 3.1. Since it is defined as Fréchet-derivative of $\tilde{J}(q)$ and is real-valued we get the *shape gradient* similarly with the gradient definition from Appendix B.3 item (iv) for $X \subset W^{k,\infty}(\mathbb{R}^d)^d$

$$\langle \nabla J, \delta q\rangle_{X^*,X} = \mathrm{d}\tilde{J}(q;\delta q) \quad \forall \delta q \in X\,. \tag{3.24}$$

Shape derivatives with the speed method

The speed method is another scheme to introduce shape calculus. It is described in [Sokolowski and Zolésio, 1992] and uses similar ideas to perturbation of identity method. We mention it because it is also one of the fundamental concepts of shape optimization and the reference above holds examples of shape derivatives for many problems. To be able to use shape derivatives from either one of these methods, we use [Delfour and Zolésio, 2011] where it is shown that under mild assumptions both yield the same derivatives. We briefly introduce the speed method to state the important Hadamard boundary representation for shape derivatives.

Speed method The speed method differs from perturbations of identity $id + q$ in the transformations $\tau(q)$ which also accept non-autonomous perturbations:

Definition 3.12. Let Ω be a bounded domain in $\mathbb{R}^d$ with a piecewise smooth boundary $\partial\Omega$ of class $\mathcal{C}^k$, $k > 0$, and material points $x \in \Omega$, where we suppose that the normal field $n(x)$ exists almost everywhere on the boundary (see also the remark in classifications of domains on page 10). For a vector field $q \in \mathcal{C}\big([0,\varepsilon);\mathcal{C}^k(\bar{\Omega};\mathbb{R}^d)\big)$ which satisfies

- $q(t,x)\cdot n(x) = 0$ for a.e. $x \in \partial\Omega$,
- $q(t,x) = 0$ at points x of $\partial\Omega$, where $n(x)$ is undefined,

the transformation τ is given as solution to the ordinary differential equation

$$\mathrm{d}_t\tau(t,x) = q(t,\tau(t,x)), \quad \tau(0,x) = x\,. \tag{3.25}$$

This definition of "perturbations" by the speed method can be found in Definition 2.17, [Sokolowski and Zolésio, 1992], where the authors also show that then there exists $0 < \delta \leq \varepsilon$ such that $\tau(t,\cdot)$ is one-to-one for each $t < \delta$.

They introduce the *Eularian derivative* of shape functionals $J(\Omega)$ in the direction of a vector field q from Definition 3.12 as

$$\mathrm{d}J(\Omega;q) := \lim_{t\downarrow 0} \frac{1}{t}\Big(J(\tau(t,\Omega)) - J(\Omega)\Big)\,. \tag{3.26}$$

Definition 3.13. Shape functionals J are shape differentiable if the Eularian derivative exists for all directions q and the mapping $q \to \mathrm{d}J(\Omega; q)$ is linear and continuous from $\mathcal{C}([0, \varepsilon); \mathcal{C}^k(\bar{\Omega}; \mathbb{R}^d))$ into $\mathbb{R}$.

Therefore, after a comparison with Definition 3.8 and (3.13) of the shape derivative from perturbation of identity, we draw the conclusion that the vector fields $q(\cdot, x)$ responsible for a local variation of the current domain Ω need not to be affine-linear anymore. However, in case of autonomous q the speed method compared to perturbation of identity requires only Gateaux- instead of Fréchet-differentiability.

A major emphasis of the speed method lies on the representation of the shape derivatives in a boundary form – a result of the Hadamard structure theorem.

Hadamard Structure Theorem In Example 3.1 let Ω be also bounded and locally Lipschitz, then the divergence theorem holds and we have a boundary representation of the shape derivative obtained by perturbation of identity

$$\mathrm{d}J(\Omega; \delta q) = \int_{\Omega} \operatorname{div}(\phi \delta q)\, \mathrm{d}x = \int_{\partial\Omega} n \cdot \delta q \phi\,. \tag{3.27}$$

This is expressed more generally by the Hadamard structure theorem which is derived with the speed method and stated in [Delfour and Zolésio, 2011, Ch. 9 Thm. 3.6]. In principle, it says that the gradient of J can be expressed as boundary representation, i.e. that the Gauß' divergence theorem holds.

Theorem 3.14 (Hadamard Structure Theorem).
Let $J(\cdot)$ be a shape functional which is shape differentiable after the speed method fashion at every domain Ω of class $\mathcal{C}^k$, Ω a subset of the holding-all D. There exists a scalar distribution g such that for a specific class of shape functionals it is an integrable function on $\partial\Omega$. If this is the case then

$$\mathrm{d}J(\Omega; \delta q) = \int_{\partial\Omega} g \delta q(0, s) \cdot n(s)\, \mathrm{d}s\,. \tag{3.28}$$

Moreover, g is related to the shape gradient ∇J in (3.24) by a trace operator on $\partial\Omega$.

The boundary representation is favorable for two reasons:

- If the structure theorem holds, we often have for the derivative of $J(\Omega) = \int_{\Omega} \phi(\Omega)\, \mathrm{d}x$ a simpler structure

$$\mathrm{d}J(\Omega; \delta q) = \int_{\Omega} \operatorname{div}\big(\phi(\Omega)(x) \delta q(x)\big)\, \mathrm{d}x = \int_{\partial\Omega} \phi(\Omega) \delta q(s) \cdot n(s)\, \mathrm{d}s\,.$$

  Then we do not have to compute any derivatives of $\phi(\Omega)$ for $\operatorname{div}\big(\phi(\Omega)(x)\delta q(x)\big)$ anymore and it may be more easily implemented.
- The boundary integral may be more efficiently evaluated than the domain integral, for instance needing less quadrature points.

However, there also is a drawback:

Remark 3.15. It has been observed, [Delfour and Zolésio, 2011, Chapter 10, Remark 2.3], that the boundary representation of the shape derivatives is often numerically

unreliable because the finite element solution of the state equation does not yield the smoothness necessary for the Hadamard structure theorem, i.e. g may not be in L^1 then.

The last remark can be seen as an argument for isogeometric analysis as it can often raise the continuity of geometry and solution representations by using smoother B-splines.

Remark 3.16. Note, that in perturbation of identity the perturbations q are elements from $W^{k,\infty}(\mathbb{R}^d)^d$ whereas in the speed method they are from $\mathcal{C}^k(\bar{\Omega})$. With Lemma B.15 functions in $W^{1,\infty}(\Omega)^d$ can be identified with continuous versions, which relates the two vector fields from speed and perturbation method.

We now move on to already regard IGA traits and use a transformation approach to express the SOP over a reference domain of fixed shape. This leads finally to the Lagrange approach to obtain first order optimality conditions suitable for an isogeometric discretization later in Chapter 4.

3.3 Transformation Approach for Isogeometric Shape Optimization

This is the essential part of this chapter and important to the thesis, so we summarize the previous sections and motivate the next steps before going into details.

Previously In the sections before we introduced perturbation of identity as a means to define derivatives w.r.t. shapes. We assumed that it yields the same sensitivities as the speed method, briefly sketched there, too. Then a notion of differentiability was established by locally varying the current domain $\Omega \in \mathcal{O}$ using perturbations of identity $\tau = id + q$ where $q \in W^{k,\infty}(\mathbb{R}^d)^d$ is the vector field affecting the change in the domain. The metric induced by $W^{k,\infty}(\mathbb{R}^d)^d$ plays a minor role from now on. Significant, though, is that the shape derivatives are displayed in terms of each current domain Ω, e.g. in the Hadamard structure

$$\mathrm{d}J(\Omega;\delta q) = \int_{\partial\Omega} \phi(\Omega)\delta q(s)\cdot n(s)\,\mathrm{d}s\,. \tag{3.29}$$

Since our aim is to tailor perturbation of identity towards IGA we filter important characteristics for our next steps.

Next steps In IGA we have a parameterization of the current domain $\Omega = G(\hat{\Omega})$ over the fixed parameter domain $\hat{\Omega}$. This leads to a shape derivative recipe for isogeometric analysis

1. Analyze the infinite-dimensional problem, e.g. answer if it is well-posed and obtain the shape derivative for the infinite-dimensional problem from perturbation of identity or the speed method in the boundary representation (3.29).
2. Apply a change of variables in the integral of the shape derivative with the isogeometric parameterization of the domain or the boundary

$$\mathrm{d}J(\Omega;\delta q) = \int_{\partial\hat{\Omega}} (\phi(\Omega)(s)\delta q(s)\cdot n(s))\circ G(\hat{s})|\det \mathrm{J}_G|\,\mathrm{d}\hat{s}\,. \tag{3.30}$$

In the publications on IGA shape optimization using the *optimize first–discretize then* approach, [Blanchard et al., 2013, Bandara et al., 2015], this method is used. As it stands, it means that

- the current domain Ω is disturbed to $(id + q)(\Omega)$,
- all SOP terms are pulled back to Ω by a change of variables to derive the shape derivatives,
- then IGA is applied which means another change of variables to pull back all terms to the parameter domain $\hat{\Omega}$.

It implies the assumption that each domain $\Omega \in \mathcal{O}$ can be given by a B-spline representation G. Our aim is to combine the two change of variables into one transformation to state the KKT optimality conditions for a parameter problem. This emphasizes a discretization of state and control through IGA and the assumption that there are such geometry representations is more pronounced.

We first review the recipe to see where IGA enters and to motivate the proposed transformation approach which we introduce subsequently. Finally, on this basis, we formulate the KKT optimality system which is discretized in Chapter 4 with isogeometric analysis.

3.3.1 Isogeometric shape sensitivities

We study the "recipe" of calculating shape sensitivities in IGA using pre-calculated shape derivatives from perturbation of identity or from the speed method. This approach shows where isogeometric discretization assumptions enter and what it signifies for our abstract frame. It is based on our results in [Fußeder et al., 2015].

For isogeometric analysis we assume that the domain $\Omega = G(\hat{\Omega})$ is parameterized by a geometry function G over the parameter domain $\hat{\Omega} = (0,1)^d$, the unit hypercube. The geometry function is supposed to satisfy either (2.31) or (2.32), i.e. it is a $\mathcal{C}^1$-diffeomorphism or bi-Lipschitz homeomorphism. We set $\hat{\mathcal{Q}} := W^{1,\infty}(\hat{\Omega})^d$

Lemma 3.17. *For a shape functional J with $j_1(u) \in L^1(\Omega)$ and $j_2(u) \in L^1(\Gamma)$,*

$$J(\Omega, u) = \int_\Omega j_1(u)(x)\,\mathrm{d}x + \int_{\Gamma \subset \partial\Omega} j_2(u)(s)\,\mathrm{d}s\,, \tag{3.31}$$

we obtain the isogeometric shape sensitivities in direction $\delta\hat{q} \in \hat{\mathcal{Q}}$ as

$$\begin{aligned} \mathrm{d}J(G;\delta\hat{q}) = &\int_{\hat{\Omega}} (\nabla j_1 \circ G \cdot \delta\hat{q} + j_1 \circ G \,\mathrm{tr}(\mathrm{J}_G^{-1}\,\mathrm{D}\delta\hat{q})|\det \mathrm{J}_G|\,\mathrm{d}\hat{x} \qquad (3.32)\\ &+ \int_{\hat{\Gamma}} \Big(\nabla j_2 \circ G \cdot \delta\hat{q}|\mathrm{J}_G^{-\mathsf{T}}\,\hat{n}| + j_2 \circ G\,\mathrm{tr}(\mathrm{J}_G^{-1}\,\mathrm{D}\delta\hat{q}) \\ &\qquad - j_2 \circ G \frac{\hat{n}^\mathsf{T}\,\mathrm{J}_G^{-1}\,\mathrm{J}_G^{-\mathsf{T}}\,\mathrm{D}\delta\hat{q}^\mathsf{T}\,\mathrm{J}_G^{-\mathsf{T}}\,\hat{n}}{|\mathrm{J}_G^{-\mathsf{T}}\,\hat{n}|}\Big)|\det \mathrm{J}_G|\,\mathrm{d}\hat{s}\,. \qquad (3.33) \end{aligned}$$

Proof using perturbation of identity and transformation rule. One considers the classical shape derivatives given in (3.21) and (3.23)

$$\mathrm{d}J(\Omega;\delta q) = \int_\Omega \nabla j_1 \cdot \delta q + j_1 \,\mathrm{div}\,\delta q\,\mathrm{d}x + \int_{\partial\Omega} \nabla j_2 \cdot \delta q + j_2(\mathrm{div}\,\delta q - n^\mathsf{T}\,\mathrm{J}_{\delta q}^\mathsf{T}\,n)\,\mathrm{d}s$$

and applies the transformation rule for $\Omega = G(\hat{\Omega})$. From Corollary 2.18 and Remark 2.19 we know that all $\delta q \in W^{1,\infty}(\mathbb{R}^d)^d$ are transported to $\hat{\mathcal{Q}} = W^{1,\infty}(\hat{\Omega})^d$. Therefore we set $\delta q \circ G =: \delta\hat{q}$. Furthermore, $(\operatorname{div} \delta q) \circ G = \operatorname{tr}(\mathrm{J}_G^{-1}\, \mathrm{D}\delta\hat{q})$ and $n \circ G = \frac{\mathrm{J}_G^{-\mathsf{T}} \hat{n}}{|\mathrm{J}_G^{-\mathsf{T}} \hat{n}|}$. □

The advantage of this view is that we can immediately reuse existing shape derivatives from the *optimize first* approach for perturbation of identity or the speed method. For shape derivatives in the Hadamard structure (3.28) it is especially beneficiary. Then all one has to do is to apply the isogeometric transformation on the boundary. This is done for IGA in [Bandara et al., 2015] and [Blanchard et al., 2013].

However, to be able to use the Hadamard structure theorem in the first place, the state needs a higher regularity. With Remark 3.15, it is therefore often computationally better to use domain integral representations of the shape gradient. This in turn implies for IGA the calculation of derivatives in the bilinear and linear form, which might prove more difficult to execute explicitly. Therefore, we propose a more direct route, i.e. combining the change of variables from perturbation of identity and from IGA.

Lemma 3.17 invokes the isogeometric paradigm for the shape variations when we chose δq as the push-forward of the parameter domain version $\delta q = \delta\hat{q} \circ G^{-1}$ and the perturbation of identity is transformed to

$$\tau(\Omega) = (id + t\delta q)(\Omega) = (id + t\delta q)(G(\hat{\Omega})) = (G + t\delta q \circ G)(\hat{\Omega})\,. \tag{3.34}$$

All in all, we learn that $\mathcal{O}$ and $W^{1,\infty}(\mathbb{R}^d)^d$ are approximated by splines, and a variation of the domain Ω is in IGA just a variation of the geometry function G as is illustrated in Figure 3.2.

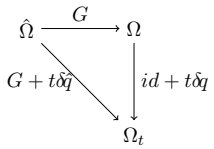


Figure 3.2: Perturbation of identity with IGA where $\delta\hat{q} = \delta q \circ G$

We take this last remark as a guidance for our transformation approach.

3.3.2 Transformed shape optimization problem

This section's aim is a shape optimization problem formulated over a fixed reference domain and fixed function spaces for the PDE, with the help of a transformation approach. We discuss our approach in connection with similar methods at the end of this section.

Transformed problem

In Definition 3.2 of abstract SOPs over a set of domains we have

$$\begin{aligned} J\colon &\{(\Omega, u)\colon \Omega \in \mathcal{O}_{\mathrm{ad}},\ u \in \mathcal{U}(\Omega)\} \to \mathbb{R} \\ \mathrm{e}\colon &\{(\Omega, u)\colon \Omega \in \mathcal{O}_{\mathrm{ad}},\ u \in \mathcal{U}(\Omega)\} \to \{z \in \mathcal{Z}(\Omega),\ \Omega \in \mathcal{O}_{\mathrm{ad}}\} \end{aligned} \tag{3.35}$$

Let $\hat{\mathcal{Q}} = \{\hat{q}\colon \hat{\Omega} \to \mathbb{R}^d\} = W^{k,\infty}(\hat{\Omega})^d$ with $\hat{\Omega} = (0,1)^d$ be a Banach space with subsets $\mathcal{G}$, $\mathcal{G}_{\text{ad}} \subset \hat{\mathcal{Q}}$ where

$$\mathcal{G}_{\text{ad}} \subset \mathcal{G} = \{G \in \hat{\mathcal{Q}}\colon G \text{ is one-to-one from } \hat{\Omega} \text{ to } G(\hat{\Omega}) \text{ and satisfies (2.31)}\}\,. \tag{3.36}$$

Alternatively, let $G \in \mathcal{G}$ satisfy (2.32). This means, the transformations in $\mathcal{G}$ can send the variational state equation from Ω to an equivalent formulation over the unit hypercube. Assume that the admissible shapes $\mathcal{O}_{\text{ad}}$ are generated by $\mathcal{G}$

$$\mathcal{O} = \{\Omega = G(\hat{\Omega})\colon G \in \mathcal{G}\} \text{ and } \mathcal{O}_{\text{ad}} = \{\Omega = G(\hat{\Omega})\colon G \in \mathcal{G}_{\text{ad}}\}\,. \tag{3.37}$$

Then we can substitute $G(\hat{\Omega})$ for Ω in (3.35) and further formulate an SOP on $\mathcal{G}_{\text{ad}}$ by using $\mathcal{U}(G) := \mathcal{U}(G(\hat{\Omega})) = \mathcal{U}(\Omega)$ and $\mathcal{Z}(G) := \mathcal{Z}(G(\hat{\Omega}))$ in

$$\begin{aligned} &J\colon \{(G,u)\colon G \in \mathcal{G}_{\text{ad}},\ u \in \mathcal{U}(G)\} \to \mathbb{R}\,,\\ &\mathrm{e}\colon \{(G,u)\colon G \in \mathcal{G}_{\text{ad}},\ u \in \mathcal{U}(G)\} \to \{z \in \mathcal{Z}(G),\ G \in \mathcal{G}_{\text{ad}}\}\,. \end{aligned} \tag{3.38}$$

We spared ourself the task of defining extra J and $\mathrm{e}(\cdot,\cdot)$ over $\mathcal{G}_{\text{ad}}$ instead of $\mathcal{O}_{\text{ad}}$. However, now we pull back J and $\mathrm{e}(\cdot,\cdot)$ to the parameter domain $\hat{\Omega}$ by a change of variables. For that, we fix the Banach spaces $\hat{\mathcal{U}} := \mathcal{U}(\hat{\Omega})$, $\hat{\mathcal{Z}} := \mathcal{Z}(\hat{\Omega})$ and finally, define new operators and function spaces over the parameter domain

$$\begin{aligned} &\hat{J}\colon \{(G,\hat{u}) \in \mathcal{G}_{\text{ad}} \times \hat{\mathcal{U}}\} \to \mathbb{R}\,,\\ &\hat{\mathrm{e}}\colon \{(G,\hat{u}) \in \mathcal{G}_{\text{ad}} \times \hat{\mathcal{U}}\} \to \hat{\mathcal{Z}}\,. \end{aligned} \tag{3.39}$$

In order to link (3.39) to (3.38) and thus to the original (3.35) let $\hat{\mathrm{e}}(\cdot,\cdot)$ be such that for all $G \in \mathcal{G}_{\text{ad}}$ the following holds

$$\hat{\mathrm{e}}(G,\hat{u}_G) = 0 \quad \Leftrightarrow \quad \mathrm{e}(G,u_G) = \mathrm{e}(G(\hat{\Omega}),u_G) = 0 \tag{3.40}$$

with $\hat{u}_G \in \hat{\mathcal{U}}$, $u_G \in \mathcal{U}(G)$ and $\hat{u}_G = u_G \circ G$. Then the abstract SOP over a set of domains from Definition 3.2 takes a form similar to an optimal control problem on a parameter domain with fixed function spaces in the following definition.

Definition 3.18. A transformed shape optimization problem on a parameter domain is given by

$$\begin{cases} \text{find } G^\star \in \mathcal{G}_{\text{ad}} \text{ such that} \\ \hat{J}(G^\star,\hat{u}^\star) \le \hat{J}(G,\hat{u}) \quad \forall G \in \mathcal{G}_{\text{ad}} \text{ and } \hat{\mathrm{e}}(G,\hat{u}) = 0\,. \end{cases} \tag{$\hat{\mathcal{P}}$}$$

Assumption (3.36) can be met with perturbation of identity: If we have a parameterization of the domain which generates the shapes $\mathscr{D}^k_{\Omega_0}$ in perturbation of identity, $G_0\colon \hat{\Omega} \to \Omega_0$, and G_0 satisfies (2.31) then $\mathcal{G} = \{\tau \circ G_0 = G_0 + q \circ G_0\colon \tau \in \mathscr{C}^k\}$.

Other transformation approaches

We remark on similar transformation approaches:

- Our notation above follows [Brandenburg et al., 2009] and [Lindemann, 2012] who defined the transformations as perturbations of identity $id + q$ over the reference domain Ω_0. That is an extension of the original perturbation of identity method by [Murat and Simon, 1976a], where Ω_0 was "only" used to generate the shapes $\mathcal{O} \subset \mathscr{D}^k_{\Omega_0}$ in (3.6) and to obtain a metric. For the isogeometric frame we proceed similar to it, however, our perturbations are defined over the parameter domain, and they perturb the parameterization G instead of the identity id. Moreover, as remarked in [Brandenburg et al., 2009, Chapter 4.9], FEM solvers still work for formulations over the physical domains. Therefore, the derivations from the transformed problem are pushed forward again from the reference Ω_0 to the physical domain Ω. In contrast, IGA solvers work only for formulations over the parameter domain which makes the transformation approaches particularly attractive.

- In [Eppler et al., 2007] and [Kiniger and Vexler, 2013, Kiniger, 2015] a domain is identified by a function q describing the boundary. Whereas convergence rates are the focus of [Eppler et al., 2007], in [Kiniger and Vexler, 2013, Kiniger, 2015] the aim are *a priori* error estimates for shape optimization problems. A wavelet boundary element method is used in [Eppler et al., 2007] to solve the state equation and to decouple state and control discretization for the convergence analysis. The approach from [Kiniger and Vexler, 2013, Kiniger, 2015] uses another PDE problem to obtain a geometry from the boundary description q. It would be interesting for a future investigation whether an IGA solver could substitute the wavelet solver, and an IGA geometry generation fulfill the domain PDE, thus recovering the convergence rates and *a priori* error estimates.

- The approach from [Ito et al., 2008] plays a role in our Lagrange formulation in the coming Section 3.4.

The embedding of shape optimization in an optimal control frame is useful because then standard arguments for existence and convergence can be used directly. Especially $(A4)$ in Assumptions 3.3 for the existence result is less elusive, because the shape continuity is not tracked over varying function spaces.

We next consider how to obtain the transformed problem $(\hat{\mathcal{P}})$ for given functionals with transformations in $\mathcal{G}$.

3.3.3 Transformed shape functionals

Let Ω be a domain in $\mathcal{O}$. Since by our assumption $\mathcal{O}$ is generated by the transformations in $\mathcal{G}$ there is a $G \in \mathcal{G}$ such that $\Omega = G(\hat{\Omega})$, and (2.31) is satisfied.

Simple shape functionals Let X be a Banach space and $\phi\colon \mathcal{O} \to X$ a shape functional. We consider transformed shape functionals $\hat{\phi}\colon \mathcal{G} \to \hat{X}$. If $\phi(\Omega) = \phi(x)|_\Omega$ is just the restriction to Ω then the transformed shape functional is simply $\hat{\phi} = \phi \circ G$. In cases involving integrals over Ω or $\Gamma \subset \partial\Omega$ with $j_1 \in L^1(\Omega)$ and $j_2 \in L^1(\Gamma)$

$$\phi(\Omega) = \int_\Omega j_1(x)\,\mathrm{d}x + \int_{\Gamma\subset\partial\Omega} j_2(s)\,\mathrm{d}s \tag{3.41}$$

we apply a change of variables to obtain the transformed shape functional

$$\hat{\phi}(G) = \int_{\hat{\Omega}} j_1 \circ G |\det \mathrm{J}_G| \,\mathrm{d}\hat{x} + \int_{\hat{\Gamma}} j_2 \circ G |\mathrm{J}_G^{-\mathsf{T}} \hat{n}| |\det \mathrm{J}_G| \,\mathrm{d}\hat{s}. \tag{3.42}$$

State equations Let $u \in \mathcal{Z}(\Omega)$ be the weak solution of $\mathrm{e}(\Omega, u) = 0$ which means it solves

$$\langle \mathrm{e}(\Omega, u), v \rangle_{\mathcal{Z}(\Omega)^*, \mathcal{Z}(\Omega)} = 0 \quad \forall v \in \mathcal{Z}(\Omega)\,, \tag{3.43}$$

or equally for our stationary model problems from Section 2.2 switching to the notation from (2.27) in a suitable Hilbert space $\mathcal{V}(\Omega)$

$$a(u, v) = l(v) \quad \forall v \in \mathcal{V}(\Omega)\,. \tag{3.44}$$

A change of variables is applicable for transformations G satisfying (2.31). Hence, with Corollary 2.18 the variational equation (3.44) is equivalent to

$$\hat{u} \in \hat{\mathcal{V}} \colon \hat{a}(G)(\hat{u}, \hat{v}) = \hat{l}(G)(\hat{v}) \quad \forall \hat{v} \in \hat{\mathcal{V}}\,, \tag{3.45}$$

where $\hat{\mathcal{V}} = \mathcal{V}(\hat{\Omega})$ and $\hat{u} = u \circ G$. By Lemma B.5 in the Appendix we can retrieve a PDE operator $\hat{\mathrm{e}}$ for $(\hat{\mathcal{P}})$ from $\hat{\mathcal{V}} \to \hat{\mathcal{V}}^*$.

Cost functionals Similarly, a cost functional can be transformed with $G \in \mathcal{G}$ so that $J(\Omega, u) = \hat{J}(G, \hat{u})$. Let the original cost have the form (3.31)

$$J(\Omega, u) = \int_{\Omega} j_1(u)(x) \,\mathrm{d}x + \int_{\Gamma \subset \partial\Omega} j_2(u)(s) \,\mathrm{d}s \tag{3.46}$$

where $j_1(u) \in L^1(\Omega)$ and $j_2(u) \in L^1(\partial\Omega)$. Then the cost functional is transformed under G from Ω to $\hat{\Omega}$

$$\begin{aligned} &J(\Omega, u) = \hat{J}(G, \hat{u}) := \\ &= \int_{\hat{\Omega}} \hat{j}_1(\hat{u}) |\det \mathrm{J}_G| \,\mathrm{d}\hat{x} + \int_{\hat{\Gamma}} \hat{j}_2(\hat{u}) |\mathrm{J}_G^{-\mathsf{T}} \hat{n}| |\det \mathrm{J}_G| \,\mathrm{d}\hat{s}\,. \end{aligned} \tag{3.47}$$

Note that in case the functions j_1 and j_2 contain a differential operator D applied to u, we additionally have to consider the transformation $\mathrm{D}u \circ G = \mathrm{D}(u \circ G)\,\mathrm{D}G^{-1}$, as in the case for bilinear and linear forms in equation (2.35)–(2.37). To take this into account we introduce the notations $\hat{j}_i = j_i \circ G$, $i = 1, 2$ in (3.47).

3.3.4 Sensitivity analysis

We proceed like in Section 3.2 for perturbation of identity, the difference being that a perturbation has not the form $\tau = id + q$ with $q \in W^{k,\infty}(\mathbb{R}^d)^d$ anymore, but $G + \hat{q}$ with $G \in \mathcal{G}$ and $\hat{q} \in \hat{\mathcal{Q}} = W^{k,\infty}(\hat{\Omega})^d$.

Definition 3.19. Let X be a Banach space and $\phi\colon \mathcal{O} \to X$ a shape functional. Then we define the *isogeometric shape derivative* of the transformed functional $\hat{\phi}$ at $G \in \mathcal{G}$ as the Fréchet-derivative of the map

$$\check{\phi}\colon \hat{\mathcal{Q}} \to X, \quad \check{\phi}(\hat{q}) := \hat{\phi}(G + \hat{q})$$

at 0. That is, $\hat{\phi}'(G) := \check{\phi}'(0)$ and we evaluate the derivatives in the direction $\delta\hat{q} \in \hat{\mathcal{Q}}$ as

$$\mathrm{d}\hat{\phi}(G; \delta\hat{q}) = \mathrm{d}_t \hat{\phi}(G + t\delta\hat{q})|_{t=0}\,. \tag{3.48}$$

Note, that we introduced the additional notation $\check{\phi}$ in Definition 3.19 to mimick perturbation of identity and because $G \in \mathcal{G} \subset \hat{\mathcal{Q}}$ but $\hat{q} \in \hat{\mathcal{Q}}$ might not be an element in $\mathcal{G}$.

As a sidetrack, we check the connection to classical perturbation of identity derivatives.

Connection to perturbation of identity The connection to perturbation of identity is given by the next lemma.

Lemma 3.20. *Let X be a Banach space and $\phi\colon \mathcal{O} \to X$ a shape functional with its transformed counter part $\hat{\phi}\colon \mathcal{G} \to X$ with $\hat{\phi}(G) = \phi(G(\hat{\Omega}))$ and $\mathcal{O} = \{\Omega = G(\hat{\Omega})\colon G \in \mathcal{G}\}$. Then we have at $G \in \mathcal{G}$ with $G(\hat{\Omega}) = \Omega \in \mathcal{O}$ for all $\delta\hat{q} \in \hat{\mathcal{Q}}$*

$$\hat{\phi}'(G)\delta\hat{q} = \phi'(\Omega)(\delta\hat{q} \circ G^{-1})\,. \tag{3.49}$$

Proof. Let $F\colon \mathcal{G} \to \mathcal{O}$ with $F(G) := G(\hat{\Omega}) = \Omega$ denote a mapping from an isogeometric transformation to a physical domain. Then we have from the chain rule in Lemma 3.9 for shape derivatives by perturbation of identity that

$$(\phi \circ F)'(G)\delta\hat{q} = \phi'(\Omega)(F'(G)\delta\hat{q})\,.$$

Furthermore, $\phi \circ F(G) = \hat{\phi}(G)$ and $F'(G)\delta\hat{q} = \delta\hat{q}(\hat{\Omega})$. Using the identity $\hat{\Omega} \ni \hat{x} = G^{-1}(x)$ for $x \in \Omega$ completes the proof as then

$$\hat{\phi}'(G)\delta\hat{q}(\hat{x}) = \phi'(\Omega)(\delta\hat{q}(\hat{x})) = \phi'(\Omega)(\delta\hat{q} \circ G^{-1}(x))\,.$$

□

As in Section 3.2.3 on perturbation of identity we provide our shape derivatives for some important cases.

Shape sensitivities in the transformation approach We establish the shape derivatives for terms due to a change of variables in integrals with isogeometric perturbations.

Lemma 3.21. *In the following we give the shape derivatives of terms due to the transformation. We define the isogeometric perturbation of $G \in \mathcal{G}$ in a fixed direction as $G_t = G + t\delta\hat{q}$ for a variation $\delta\hat{q} \in \hat{\mathcal{Q}}$. Then it holds at G in direction $\delta\hat{q}$:*

(i) $G'\delta\hat{q} = \delta\hat{q}$.

(ii) $\hat{\phi}(G) := \mathrm{J}_G$ *is differentiable with* $\mathrm{d}\hat{\phi}(G;\delta\hat{q}) = \mathrm{J}_{\delta\hat{q}}$.

(iii) $\hat{\phi}(G) := |\det \mathrm{J}_G|$ *is differentiable with* $\mathrm{d}\hat{\phi}(G;\delta\hat{q}) = |\det \mathrm{J}_G| \operatorname{tr}(\mathrm{J}_G^{-1}\,\mathrm{J}_{\delta\hat{q}})$.

(iv) The map to the inverse Jacobian of the perturbation $\hat{\phi}(G) := \mathrm{J}_G^{-1}$ *is differentiable with* $\mathrm{d}\hat{\phi}(G;\delta\hat{q}) = -\,\mathrm{J}_G^{-1}\,\mathrm{J}_{\delta\hat{q}}\,\mathrm{J}_G^{-1}$.

(v) The transformation term $\hat{\phi}(G) := \mathrm{J}_G^{-1}\,\mathrm{J}_G^{-\mathsf{T}}$ *occuring for instance in bilinear forms is differentiable with* $\mathrm{d}\hat{\phi}(G;\delta\hat{q}) = -\,\mathrm{J}_G^{-1}(\mathrm{J}_{\delta\hat{q}}\,\mathrm{J}_G^{-1} + \mathrm{J}_G^{-\mathsf{T}}\,\mathrm{J}_{\delta\hat{q}}^{\mathsf{T}})\,\mathrm{J}_G^{-\mathsf{T}}$

Proof. In the following, we use the notation $\phi'(t) = \mathrm{d}_t\phi(t)$.

(i) $\mathrm{d}_t G_t|_{t=0} = \delta\hat{q}$

(ii) $\mathrm{d}_t\,\mathrm{J}_{G_t}|_{t=0} = \mathrm{d}_t(\mathrm{J}_G + t\,\mathrm{J}_{\delta\hat{q}})|_{t=0} = \mathrm{J}_{\delta\hat{q}}$

(iii) We use the directional derivative of the Jacobian in point (ii) above and the following derivatives

- $\mathrm{d}_t \det\phi(t)|_{t=0} = \det\phi(0)\operatorname{tr}(\phi^{-1}(0)\phi'(0))$
- $\mathrm{d}_t|\phi(t)|\big|_{t=0} = \operatorname{sign}(\phi(0))\phi'(0)$

together with the chain rule to differentiate

$$\mathrm{d}_t|\det \mathrm{J}_{G_t}|\big|_{t=0} = \operatorname{sign}(\det \mathrm{J}_G)\det \mathrm{J}_G \operatorname{tr}(\mathrm{J}_G^{-1}\,\mathrm{J}_{\delta\hat{q}}) = |\det \mathrm{J}_G|\operatorname{tr}(\mathrm{J}_G^{-1}\,\mathrm{J}_{\delta\hat{q}})\,.$$

(iv) $0 = \mathrm{d}_t I = \mathrm{d}_t(\mathrm{J}_{G_t}^{-1}\,\mathrm{J}_{G_t})|_{t=0} = \mathrm{d}_t\,\mathrm{J}_{G_t}^{-1}|_{t=0}\,\mathrm{J}_G + \mathrm{J}_G^{-1}\,\mathrm{d}_t\,\mathrm{J}_{G_t}|_{t=0} \overset{(ii)}{=}$
$= \mathrm{d}_t\,\mathrm{J}_{G_t}^{-1}|_{t=0}\,\mathrm{J}_G + \mathrm{J}_G^{-1}\,\mathrm{J}_{\delta\hat{q}}$.

(v) $\mathrm{d}_t(\mathrm{J}_{G_t}^{-1}\,\mathrm{J}_{G_t}^{-\mathsf{T}})|_{t=0} = \mathrm{d}_t\,\mathrm{J}_{G_t}^{-1}|_{t=0}\,\mathrm{J}_G^{-\mathsf{T}} + \mathrm{J}_G^{-1}\,\mathrm{d}_t\,\mathrm{J}_{G_t}^{-\mathsf{T}}|_{t=0} \overset{(iv)}{=}$
$= -\,\mathrm{J}_G^{-1}\,\mathrm{J}_{\delta\hat{q}}\,\mathrm{J}_G^{-1}\,\mathrm{J}_G^{-\mathsf{T}} - \mathrm{J}_G^{-1}\,\mathrm{J}_G^{-\mathsf{T}}\,\mathrm{J}_{\delta\hat{q}}^{\mathsf{T}}\,\mathrm{J}_G^{-\mathsf{T}}$

□

With the recurrent derivatives above we are able to give the directional derivatives of shape functionals in the transformed shape optimization problem $(\hat{\mathcal{P}})$:

Lemma 3.22. *The isogeometric shape derivatives at* $G \in \mathcal{G}$ *in a direction of* $\delta\hat{q} \in \hat{\mathcal{Q}}$ *of the integrals in* (3.42) *is given by*

$$\mathrm{d}\hat{J}(G;\delta\hat{q}) = \int_{\hat{\Omega}} (\nabla j_1 \circ G \cdot \delta\hat{q} + j_1 \circ G \operatorname{tr}(\mathrm{J}_G^{-1}\,\mathrm{J}_{\delta\hat{q}})|\det \mathrm{J}_G|\,\mathrm{d}\hat{x} \tag{3.50}$$

$$+ \int_{\hat{\Gamma}} \Big(\nabla j_2 \circ G \cdot \delta\hat{q}|\mathrm{J}_G^{-\mathsf{T}}\,\hat{n}| + j_2 \circ G \operatorname{tr}(\mathrm{J}_G^{-1}\,\mathrm{J}_{\delta\hat{q}}) - j_2 \circ G \frac{\hat{n}^{\mathsf{T}}\,\mathrm{J}_G^{-1}\,\mathrm{J}_G^{-\mathsf{T}}\,\mathrm{J}_{\delta\hat{q}}^{\mathsf{T}}\,\mathrm{J}_G^{-\mathsf{T}}\,\hat{n}}{|\mathrm{J}_G^{-\mathsf{T}}\,\hat{n}|}\Big)|\det \mathrm{J}_G|\,\mathrm{d}\hat{s}. \tag{3.51}$$

Proof. We use the chain rule to differentiate $j_1 \circ G$, $j_2 \circ G$, and utilize Lemma 3.21 to find the directional derivatives of $|\det \mathrm{J}_G|$. We note furthermore that for $\phi\colon \mathbb{R} \to \mathbb{R}^d$, $\phi(t) = \begin{pmatrix}\phi_1 & \dots & \phi_d\end{pmatrix}^\mathsf{T} \in H^1(\mathbb{R})^d$ we have $|\phi(t)| = \sqrt{\sum_{i=1}^d \phi_i(t)^2}$ and

$$\mathrm{d}_t|\phi(t)| = \frac{\sum_{i=1}^d \phi_i(t)\,\mathrm{d}_t\phi_i(t)}{|\phi(t)|} = \frac{\phi(t)\cdot \mathrm{d}_t\phi(t)}{|\phi(t)|}\,.$$

Thus, setting $\phi(t) = \mathrm{J}_{G_t}^{-\mathsf{T}}\,\hat{n}$ we find with Lemma 3.21 (iv)

$$\mathrm{d}_t|\mathrm{J}_{G_t}^{-\mathsf{T}}\,\hat{n}|\Big|_{t=0} = \frac{\mathrm{J}_G\,\hat{n}\cdot(-\mathrm{J}_G^{-\mathsf{T}}\,\mathrm{J}_{\delta\hat{q}}^{\mathsf{T}}\,\mathrm{J}_G^{-\mathsf{T}})\hat{n}}{|\mathrm{J}_{G_t}^{-\mathsf{T}}\,\hat{n}|}\,.$$

□

Remark 3.23. In case j_i, $i = 1, 2$, depend on the state u and thus implicitly on a domain Ω, we have that $u \circ G = \hat{u}$. We use then the notation $\hat{j}_i$ from (3.3.3) and in the presence of differential operators acting on u, the transformation $\hat{j}_i = j_i(u) \circ G$ contains operations with G like the inverse Jacobian. This has to be differentiated then, too.

We use these sensitivities to derive the KKT optimality system in the next section.

3.4 Lagrange Formalism for Isogeometric Shape Optimization

In this section we introduce a Lagrange formalism for shape optimization considering isogeometric analysis traits to obtain an isogeometric suitable optimality system. Instead of problem $(\mathcal{P})$ we formulate the optimality system for the parameter problem $(\hat{\mathcal{P}})$.

We consider the Lagrangian

$$\begin{aligned}&\mathcal{L}\colon \mathcal{G}_{\mathrm{ad}} \times \hat{\mathcal{V}} \times \hat{\mathcal{V}} \to \mathbb{R}\\ &\mathcal{L}(G,\hat{u},\hat{z}) = \hat{J}(G,\hat{u}) + \hat{l}(G)(\hat{z}) - \hat{a}(G)(\hat{u},\hat{z})\end{aligned} \tag{3.52}$$

and with it, get an optimality system since stationary points of $\mathcal{L}$ are candidates of local minimums of the shape optimization problem. The KKT system from Theorem 2.29 uses in the case of an SOP over parameter spaces the following isogeometric sensitivities:

- $\mathrm{d}_{\hat{z}}\mathcal{L}(G,\hat{u},\hat{z};\delta\hat{z}) = \hat{l}(G)(\delta\hat{z}) - \hat{a}(G)(\hat{u},\delta\hat{z})$
- $\mathrm{d}_{\hat{u}}\mathcal{L}(G,\hat{u},\hat{z};\delta\hat{u}) = \langle \hat{J}'_{\hat{u}}(q,\hat{u}) - \hat{a}'_{\hat{u}}(G)(\hat{u},\hat{z}), \delta\hat{u}\rangle_{\hat{\mathcal{V}}^*,\hat{\mathcal{V}}}$
- $\mathrm{d}_{\hat{q}}\mathcal{L}(G,\hat{u},\hat{z};\delta\hat{q}) = \langle \hat{J}'_{\hat{q}}(G,\hat{u}) + \hat{l}'_{\hat{q}}(G)(\hat{z}) - \hat{a}'_{\hat{q}}(G)(\hat{u},\hat{z}), \delta\hat{q}\rangle_{\hat{\mathcal{Q}}^*,\hat{\mathcal{Q}}}$

Since in the Lagrangian the variables are independend, we find the derivative of $\hat{J}$ without considering the implicit dependence of the state on the control:

Lemma 3.24. *For a shape functional J as in* (3.31) *or in* (3.46)*, we obtain the isogeometric shape sensitivities in direction $\delta\hat{q} \in \hat{\mathcal{Q}}$ as*

$$\mathrm{d}\hat{J}(G;\delta\hat{q}) = \int_{\hat{\Omega}} (\nabla j_1 \circ G \cdot \delta\hat{q} + j_1 \circ G \,\mathrm{tr}(\mathrm{J}_G^{-1}\,\mathrm{D}\delta\hat{q}))|\det \mathrm{J}_G|\,\mathrm{d}\hat{x} \tag{3.53}$$

$$+ \int_{\hat{\Gamma}} \Big(\nabla j_2 \circ G \cdot \delta\hat{q}|\mathrm{J}_G^{-\intercal}\,\hat{n}| + j_2 \circ G\,\mathrm{tr}(\mathrm{J}_G^{-1}\,\mathrm{D}\delta\hat{q}) - j_2 \circ G \frac{\hat{n}^\intercal\,\mathrm{J}_G^{-1}\,\mathrm{J}_G^{-\intercal}\,\mathrm{D}\delta\hat{q}^\intercal\,\mathrm{J}_G^{-\intercal}\,\hat{n}}{|\mathrm{J}_G^{-\intercal}\,\hat{n}|}\Big)|\det \mathrm{J}_G|\,\mathrm{d}\hat{s}. \tag{3.54}$$

Proof using isogeometric perturbations. We directly use the directional derivative on the transported shape functional (3.47)

$$J(\Omega_t, u_t) = \hat{J}(G_t, \hat{u}_t) = \int_{\hat{\Omega}} \hat{\jmath}_1(\hat{u})(\hat{x})|\det \mathrm{J}_{G_t}|\,\mathrm{d}\hat{x} + \int_{\hat{\Gamma}} \hat{\jmath}_2(\hat{u})(\hat{x})|\mathrm{J}_{G_t}^{-\intercal}\,\hat{n}||\det \mathrm{J}_{G_t}|\,\mathrm{d}\hat{s}$$

with $\hat{\jmath}_1 = j_1 \circ G_t$ and $\hat{\jmath}_2 = j_2 \circ G_t$. We now differentiate w.r.t. the parameter t and evaluate at $t = 0$: We note $\mathrm{d}\hat{\jmath}_i(0;\delta\hat{q}) = \mathrm{d}_t j_i \circ G_t|_{t=0} = \nabla j_i \circ G_0\,\mathrm{d}_t G_t|_{t=0} = \nabla j_i \circ G \cdot \delta\hat{q}$ for $i = 1, 2$ if there is no differential operator acting on $\hat{u}$. Otherwise, $\hat{\jmath}_i$ also depends on G_t, then denote $\hat{\jmath}_{i,t}$ and use the chain rule. We utilize Lemma 3.21 from which follows the rest. □

The derivatives of the transported forms $\hat{a}(\cdot)$ and $\hat{l}(\cdot)$ defined over $\mathcal{G}$ follow by the same means. If we compare the shape derivatives in Lemma 3.24 to those obtained from using the transformation formula on pre-calculated ones, we see that they formally are equivalent. However, here we have obtained them by differentiating the transformed, infinite-dimensional shape optimization problem and can discretize now the control space $\hat{\mathcal{Q}}$ by B-splines, NURBS, and NURBS with variable weights in the optimality system:

Lemma 3.25. *The first order necessary optimality system in isogeometric shape optimization translates to*

$$\hat{a}(G)(\hat{u}, \delta\hat{v}) = \hat{l}(G)(\delta\hat{v}) \qquad \forall \delta\hat{v} \in \hat{\mathcal{V}} \qquad \textit{state equation,} \tag{3.55}$$

$$\mathrm{d}_{\hat{u}}\hat{J}(G, \hat{u}; \delta\hat{u}) = \hat{a}(G)(\delta\hat{u}, \hat{z}) \quad \forall \delta\hat{u} \in \hat{\mathcal{V}} \qquad \textit{adjoint equation,} \tag{3.56}$$

with $\mathrm{d}_{\hat{u}}\hat{J}(G, \hat{u}; \delta\hat{u}) = \mathrm{d}_t\hat{J}(G, \hat{u} + t\delta\hat{u})|_{t=0}$. The shape gradient is formed by directional derivatives

$$\mathrm{d}_{\hat{q}}\mathcal{L}(G, \hat{u}, \hat{v}; \delta\hat{q}) = \mathrm{d}_{\hat{q}}\hat{J}(G, \hat{u}; \delta\hat{q}) + \mathrm{d}_{\hat{q}}\hat{l}(\hat{v}; \delta\hat{q}) - \mathrm{d}_{\hat{q}}\hat{a}(\hat{u}, \hat{v}; \delta\hat{q}) \tag{3.57}$$

with $\mathrm{d}_{\hat{q}}\hat{J}(G, \hat{u}; \delta\hat{q})$ given by Lemma 3.24,

$$\mathrm{d}_{\hat{q}}\hat{l}(\hat{v}; \delta\hat{q}) = \mathrm{d}_t\hat{l}(G + t\delta\hat{q})(\hat{v})|_{t=0} \textit{ and } \mathrm{d}_{\hat{q}}\hat{a}(G)(\hat{u}, \hat{v}; \delta\hat{q}) = \mathrm{d}_t\hat{a}(G + t\delta\hat{q})(\hat{u}, \hat{v})|_{t=0}.$$

A Lagrange approach to shape optimization is by no means a feature of isogeometric analysis solely, and we have to consider the same as does classical shape optimization: As noted on page 27 the Lagrange approach circumvents the derivative of the state w.r.t.

the control. While [Allaire, 2007, p. 151] cautions that this method is a convenient albeit formal one, because it presumes shape differentiability of the state, the more recent monograph on the Lagrange method [Ito and Kunisch, 2008] shows that shape differentiability of the state u can often be waived. The Lagrange method is applicable then, and not only formal, if either

- shape differentiablility in the classical sense, including that of the state, can be shown, or
- if the (mild) assumptions given in [Ito and Kunisch, 2008] hold.

4 Discretization in Isogeometric Analysis

B-splines and NURBS are at the heart of isogeometric analysis, be it as basis for designing geometries, approximating simulation results, or controlling shapes in optimization. They allow for a quick evaluation due to a polynomial character, and flexibility because of a piecewise definition. We introduce B-splines together with some relevant properties in **Section 4.1** and use them in the subsequent sections to show their use for modeling, analysis and shape calculus. So, B-spline spaces are approximation spaces for geometries in **Section 4.2**, serve as ansatz spaces for PDEs in **Section 4.3** and furthermore define perturbations of domains in **Section 4.4**.

4.1 B-splines, NURBS and Polynomial Spaces

This section introduces B-splines and NURBS as special piecewise and piecewise, rational polynomials respectively. The focus lies on fixing the notation and properties for their later destiny for geometry representations, PDE solutions, and shape gradients. In particular the tensor product splines are the building blocks of two- and three-dimensional bodies, and their piecewise support hands us the analogy to FEM triangulations. As locally defined test functions they constitute the isogeometric finite elements. Form optimization in IGA then draws on both geometric and function space concepts to define gradients with respect to shapes. For more background and features of B-splines we refer to various sources like [Schumaker, 1981, De Boor, 2001]. Also, a more elaborate treatment of NURBS and especially numerical algorithms for computing points on a geometry, efficient evaluations of basis functions, etc., are left out in our introduction, but can be found in the monograph [Piegl and Tiller, 1995].

We begin this section with piecewise polynomials because they characterize B-splines and, moreover, they define finite elements in Section 4.3.1. From there, we move to B-splines and in a second step to their rational versions NURBS. After recapitulating some important properties, we review how to obtain refined or compatible spaces: The refinement is needed to catch features of geometries and approximate solutions of PDEs better. We introduce a compatibility criterion for two different spline spaces in order to realize computationally the separate discretization of shapes and of the state in shape optimization later.

4.1.1 Polynomial Spaces

We follow [De Boor, 2001, p. 1] to say that a *polynomial of degree* p is given by $\pi(x) = a_0 + a_1 x + \ldots + a_p x^p$. It has $p+1$ degrees of freedom, its coefficients a_i, and therefore is of *order* $(p+1)$. We call $\mathcal{P}_p(\Omega)$ the space of all polynomials on $\Omega \subset \mathbb{R}$ of degree less than or equal to p.

Multivariate polynomials are defined as products of univariate polynomials. That is for $\Omega \subset \mathbb{R}^d$ where $d \geq 1$ we have in the multi-index notation, $\alpha = (\alpha_1, \ldots, \alpha_d)$, the

following examples of multivariate polynomials of degree p

$$\mathcal{P}_{T,p}(\Omega) := \left\{ \pi(x) = \sum_{|\alpha|\leq p} a_\alpha x^\alpha = \sum_{|\alpha|\leq p} a_{\alpha_1\cdots\alpha_d} x_1^{\alpha_1}\cdots x_d^{\alpha_d} \text{ for } x\in\Omega \right\} \tag{4.1}$$

and $\dim \mathcal{P}_{T,p}(\Omega) = \binom{d+p}{p}$. Similarly, we need the polynomial space

$$\mathcal{P}_{Q,p}(\Omega) := \left\{ \pi(x) = \sum_{\alpha_i\leq p,\, 1\leq i\leq d} a_{\alpha_1\cdots\alpha_d} x_1^{\alpha_1}\cdots x_d^{\alpha_d} \text{ for } x\in\Omega \right\} \tag{4.2}$$

for a local finite element basis with $\dim \mathcal{P}_{Q,p}(\Omega) = (p+1)^d$. We have the inclusion $\mathcal{P}_{T,p} \subset \mathcal{P}_{Q,p} \subset \mathcal{P}_{T,dp}$, [Ciarlet, 1979, p. 56].

We now define the piecewise polynomials.

Definition 4.1. On a sequence of N *break points* $\Xi := (\upsilon_1, \ldots, \upsilon_N)$ with strictly increasing points $\upsilon_j \in \mathbb{R}$ we denote $Q_j := (\upsilon_j, \upsilon_{j+1})$ and set $\Omega = \cup \bar{Q}_j$. A *piecewise polynomial* π of degree p is characterized by $\pi|_{Q_j} =: \pi_j$ with $\pi_j \in \mathcal{P}_p(Q_j)$ for all $j = 1, \ldots, N-1$.

For such a π to be globally k-times continuously differentiable, $\pi \in \mathcal{C}^k(\bar{\Omega})$, we make it first available at the boundary break points. That is, we extrapolate $\pi(\upsilon_1) := \pi_1(\upsilon_1)$ and $\pi(\upsilon_N) := \pi_{N-1}(\upsilon_N)$. Second, the derivatives $\mathrm{d}^l\pi$ must exist for all $l = 0, \ldots, k$ also at the break points.

In [Schumaker, 1981, p. 5] *polynomial splines* of degree p are $p-1$ times continuously differentiable piecewise polynomials π of degree p, which we pursue next.

4.1.2 B-splines

B-splines are piecewise polynomials with a local nature which allow a quick evaluation. They are also flexible since changes in one parameter affects only a part and not the whole of the B-spline. Here, we define them via the Cox-De Boor recursive formula.

Definition 4.2. Given a *knot vector* $\Xi := (\xi_1, \ldots, \xi_{n+p+1})$ with *knots* $\xi_i \in [0,1]$ and $\xi_i \leq \xi_{i+1}$, a *B-spline function of degree* p is defined recursively

$$\text{for } p = 0\text{:} \quad N_{i,0}(\hat{x}) = \begin{cases} 1 \text{ if } \xi_i \leq \hat{x} < \xi_{i+1} \\ 0 \text{ otherwise} \end{cases} \tag{4.3}$$

$$\text{for } p > 0\text{:} \quad N_{i,p}(\hat{x}) = \frac{\hat{x}-\xi_i}{\xi_{i+p}-\xi_i} N_{i,p-1}(\hat{x}) + \frac{\xi_{i+p+1}-\hat{x}}{\xi_{i+p+1}-\xi_{i+1}} N_{i+1,p-1}(\hat{x}) \tag{4.4}$$

for an $\hat{x} \in [0,1]$ and $i = 1, \ldots, n$, with the convention $^0/_0$ to be zero. The recurrence formula is modified such that $N_{n,p}$ is defined also for $\hat{x} = \xi_{n+1}$ to allow for a definition of B-splines over the full interval of $[0,1]$ instead of only $[0,1)$. We call the interval between two successive knots (ξ_i, ξ_{i+1}) the i-th *knot span*. The order of $N_{i,p}$ is $p+1$.

Remark 4.3. In [Piegl and Tiller, 1995, p. 50] the recurrence formula is defined for knots in $\mathbb{R}$. We restricted them to the interval $[0,1]$ for our isogeometric purposes later.

These univariate B-splines are linearly independent for $i = 1, \dots, n$ and form a basis $\mathcal{B}(\Xi, p) := \{N_{i,p} \colon i = 1, \dots, n\}$ which generates the univariate *spline space*

$$\mathcal{S}(\Xi, p) := \operatorname{span} \mathcal{B}(\Xi, p). \tag{4.5}$$

If we confine ourselves to $\Xi := (0, \dots, 0, \xi_{p+2}, \dots, \xi_n, 1, \dots, 1)$, that is we use only *open knot vectors* where the first and last knot each occur $p + 1$ times, we enforce $N_{1,p}(0) = 1 = N_{n,p}(1)$. An example of such a basis is shown in Figure 4.1 where we have $n = 8$ B-spline basis functions of degree $p = 2$ from an open knot vector with 11 knots.

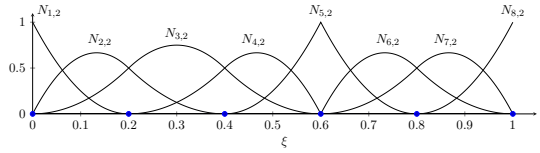


Figure 4.1: B-spline basis for knot vector $\Xi = (0, 0, 0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{3}{5}, \frac{4}{5}, 1, 1, 1)$

Next, we extend the spline space to multiple dimensions.

Definition 4.4. For d open knot vectors Ξ_i, $i = 1, \dots, d$, and corresponding spline spaces $\mathcal{S}_i := \mathcal{S}(\Xi_i, p_i)$ we form the *tensor product spline space* $\mathcal{S}(\Xi_1, p_1; \dots; \Xi_d, p_d)$ like [Schumaker, 1981], where

$$\begin{aligned} &\mathcal{S}(\Xi_1, p_1; \dots; \Xi_d, p_d) := \bigotimes_{i=1}^{d} \mathcal{S}_i \text{ with} \\ &n \text{ basis functions } N_i(\hat{x}) = N^{(1)}_{i_1,p_1}(\hat{x}_1) \cdots N^{(d)}_{i_d,p_d}(\hat{x}_d) \end{aligned} \tag{4.6}$$

over $\hat{x} \in [0, 1]^d$. Let n_i be the dimension of the univariate spline space $\mathcal{S}_i$, then the tensor product space $\mathcal{S}(\Xi_1, p_1; \dots; \Xi_d, p_d)$ has dimension $n = \prod_{i=1}^{d} n_i$ and a basis $\mathcal{B} := \{N_i = \prod_{j=1}^{d} N^{(j)}_{i_j,p_j}(\hat{x}_j) \colon N^{(j)}_{i_j,p_j} \in \mathcal{B}(\Xi_j, p_j)\}$.

Although B-splines can represent a large number of curves important types like circles are excluded. Therefore, we next extend the theory to rational piecewise polynomials which are better suited to conics.

4.1.3 NURBS

NURBS is an acronym that stands for "Non-Uniform Rational B-Splines" and describes weighted B-splines over non-uniformly spaced knot vectors. They are attractive to IGA because they contain rational polynomials which allows to design important geometries like disks and spheres. We give their definition and properties according to Chapter 5 in [Piegl and Tiller, 1995].

Definition 4.5. Given a univariate B-spline basis $\mathcal{B}(\Xi, p) := \{N_{i,p} \colon i = 1, \dots, n\}$ and a fixed *weight vector* $W = (\omega_i)_{i=1,\dots,n}$, $\omega_i > 0$, let

$$w(\hat{x}) := \sum_{i=1}^{n} N_{i,p}(\hat{x}) \omega_i \tag{4.7}$$

denote a positive *weight function* from $[0,1]^d \to \mathbb{R}^+$ from which we obtain a NURBS basis

$$\mathcal{B}(W,\Xi,p) := \left\{ R_{i,p} = \frac{\omega_i}{w} N_{i,p} \colon N_{i,p} \in \mathcal{B} \right\} \tag{4.8}$$

and the univariate NURBS space

$$\mathcal{S}(W,\Xi,p) := \operatorname{span} \mathcal{B}(W,\Xi,p)\,. \tag{4.9}$$

We proceed similar to Section 4.1.2 to obtain multivariate NURBS.

Definition 4.6. For d open knot vectors Ξ_i, corresponding B-spline basis $\mathcal{B}_i := \mathcal{B}(\Xi_i, p_i)$ with n_i basis functions, and fixed weight vectors W_i, $i = 1,\dots,d$, we first form a scalar weight function

$$w(\hat{x}) := \sum_{i_1=1}^{n_1} \dots \sum_{i_d=1}^{n_d} \omega_{i_1}^{(1)} N_{i_1,p_1}^{(1)}(\hat{x}_1) \cdots \omega_{i_d}^{(d)} N_{i_d,p_d}^{(d)}(\hat{x}_d)\,, \tag{4.10}$$

where $\omega_{i_j}^{(j)} \in W_j$ and $N_{i_j,p_j}^{(j)} \in \mathcal{B}_j$. Then we have the d-variate NURBS basis

$$\mathcal{B}(W_1,\Xi_1,p_1;\dots,W_d,\Xi_d,p_d) := \left\{ R_i(\hat{x}) := \frac{\prod_{j=1}^{d} \omega_{i_j}^{(j)} N_{i_j,p_j}^{(j)}(\hat{x}_j)}{w(\hat{x})} \text{ for all } i_j = 1,\dots,n_j,\ j = 1,\dots,d \right\} \tag{4.11}$$

and the tensor product like *NURBS space* is given by

$$\mathcal{S}(W_1,\Xi_1,p_1;\dots;W_d,\Xi_d,p_d) := \operatorname{span} \mathcal{B}(W_1,\Xi_1,p_1;\dots,W_d,\Xi_d,p_d)\,. \tag{4.12}$$

Remark 4.7. Due to the weight function w, a NURBS space is not a tensor product space anymore. However, we go with IGA literature and call it tensor product like.

Remark 4.8. If the polynomial degree p of the B-splines is clear from the context, we omit it in the index of the basis functions to simplify the notation. For instance in (4.6) and (4.11) we dropped it for the basis functions of the multivariate spline spaces. Also, we assume that the univariate factors N_{i_j,p_j} for each space dimension $j = 1,\dots,d$ are ordered through the tensor product (like) structure lexicographically and hence the counting index i of N_i and R_i of multivariate basis functions should be understood as the natural number resulting from this order. If we emphasize the vector space character of B-splines or NURBS to use their common properties we denote these spaces (uni- or multivariate) by $\mathcal{S}$. In this case, since univariate are but a special case of multivariate basis functions we do not distinguish them, i.e. we drop the degree index also in the univariate case and write then N_i and R_i instead of $N_{i,p}$ and $R_{i,p}$.

4.1.4 Properties of B-splines and NURBS

We summarize some of the valuable properties of B-splines and NURBS for isogeometric analysis.

Definition 4.9. If a knot span (ξ_j, ξ_{j+1}) of an open knot vector $\Xi = (\xi_1, \dots, \xi_{n+p+1})$ is not empty, i.e. $\xi_j \neq \xi_{j+1}$, it is called *element*. Define $\Upsilon := (\upsilon_1, \dots, \upsilon_N)$ as the sequence of distinct knots from Ξ, where $N \leq n + p + 1$. Then, each knot interval $Q_j := (\upsilon_j, \upsilon_{j+1})$ for $1 \leq j < N$ is an element. A knot υ_j has *multiplicity* $\mu_j > 0$ if it occurs μ_j times in Ξ. The unique knots Υ correspond to the break points of a piecewise polynomial space.

It will be convenient at times to use Υ as set which we do not denote extra.

Lemma 4.10. *With these additional definitions one observes the following properties for B-splines $N_{i,p}$ or NURBS $R_{i,p}$ from an open knot vector $\Xi = (\xi_1, \dots, \xi_{n+p+1})$.*

(i) Partition of unity: For an arbitrary knot span $[\xi_i, \xi_{i+1})$, $\sum_{j=i-p}^{i} N_{j,p} = 1$. Therefore, the weight function reduces to $w(\hat{x}) = 1$ in Definition 4.5 if all weights are set to a constant $\omega_i = a \neq 0$, and the rational NURBS basis turns into a non-rational B-spline basis.

(ii) Local support: $\operatorname{supp}(N_{i,p}) = \overline{[\xi_i, \xi_{i+p+1})} = [\xi_i, \xi_{i+p+1}]$.

(iii) Local enumeration: On the j-th knot span (ξ_j, ξ_{j+1}) live $p+1$ B-splines $N_{i,p}$ with $i = j-p, \dots, j$. Set $\tau_j := j - p$ on this knot span, then we have a local enumeration of these B-splines $N_{i,p} = N_{\tau_j+k,p}$ for $k = 0, \dots, p$. Correspondingly note, that therefore a basis function $N_{i,p}$ has local number $k = i - \tau_j$ on its knot spans $j = i, \dots, i+p+1$.

(iv) Continuity: For a B-spline function ϕ of $\mathcal{S}(\Xi, p)$ it holds that $\phi \in \mathcal{C}^\infty(Q_j)$ for all elements $j = 1, \dots, N-1$, where N is the number of unique knots in Υ. At element interfaces $\hat{x} = \upsilon_j \in \Upsilon$ the continuity is determined by the multiplicity μ_j, $\phi \in \mathcal{C}^{p-\mu_j}(\hat{x})$; in case of no repeated knots besides first and last, we say ϕ has maximal global continuity $\mathcal{C}^{p-1}$. We refer again to Figure 4.1 on page 53 where a repeated knot introduces a cusp in an otherwise $\mathcal{C}^1$ basis.

Because of the partition of unity property (i), B-splines are special cases of NURBS and properties (ii)–(iv) above hold also for NURBS.

Proof. We comment shortly on the proofs for the properties because they illuminate the nature of B-splines and NURBS; they can be found in [Piegl and Tiller, 1995].

(i) Property P2.4 in [Piegl and Tiller, 1995]. Note, that actually $w = a$, however, it cancels with the weights in the numerator.

(ii) Proof by induction over p. At $p = 0$, $\operatorname{supp}(N_{i,p}) \neq 0$ on $[\xi_i, \xi_{i+1}]$, for $p+1$ we use the recurrence formula (4.4) which gives us $\operatorname{supp}(N_{i,p+1}) = \operatorname{supp}(N_{i,p}) \cup \operatorname{supp}(N_{i+1,p})$.

(iii) From the previous point we know $\operatorname{supp}(N_{\tau_j+k,p}) = [\xi_{\tau_j+k}, \xi_{\tau_j+k+p+1}]$. Since $\tau_j + k \leq j$ and similarly, $\tau_j + k + p + 1 \geq j + 1$ for all possible k, $[\xi_j, \xi_{j+1}]$ is contained in the support.

(iv) Since the B-spline ϕ is a piecewise polynomial inside an element it is infinitely often differentiable. At element interfaces, i.e. at knots of multiplicity μ, the $\mathcal{C}^{p-\mu}$ continuity is inherited from the B-spline continuity: At a knot with multiplicity μ the B-spline basis $N_{i,p}$ is $p-\mu$ times continuously differentiable, properties P2.5 and P4.11 in [Piegl and Tiller, 1995].

□

The higher regularity of basis functions across break points or element boarders given by (iv) benefits simulation results for instance in eigenvalue problems, [Hughes et al., 2006]. The local support from (ii) makes an element-wise assembly of system matrices similar to FEM possible, e.g. in the tutorial [Vuong et al., 2010]. Moreover due to (ii), we can change a spline geometry locally which is exploited in the design process with CAGD.

To capture such local features, or on a global scale, to compare the approximation power of different splines we discuss the relation between two spline spaces next.

4.1.5 Nested and compatible spline spaces

This section is motivated by the following question. Can we modify a given spline space such that it has better approximation qualities? The answer will be helpful when it comes to refining geometries and ansatz spaces in simulation for better approximations. We also introduce compatible spaces for formulations that contain two different spline spaces: If they are compatible an implementation is straight-forward.

Definition 4.11. Two spaces X, Y are *nested* if $X \subset Y$. Especially two univariate spline spaces are nested, if for knot vectors Ξ, Ξ' and degrees p, p' the spline spaces satisfy

$$\mathcal{S}(\Xi, p) \subset \mathcal{S}(\Xi', p')\,. \tag{4.13}$$

The spline space $\mathcal{S}(\Xi', p')$ is larger and thus called *finer*, whereas $\mathcal{S}(\Xi, p)$ is smaller or *coarser*. This can be naturally extended to multivariate spline spaces.

There are two important methods to obtain nested, finer spline spaces. In the first, we increase the number of basis functions by inserting new distinct knots. The second way increases the multiplicity of knots to raise the polynomial degree of the B-spline basis.

Knot insertion

A natural way to get more basis functions is to supply more break points. That yields nested spaces which is shown with the B-spline recursion:

Lemma 4.12. *Let $\mathcal{B}(\Xi, p)$ denote a univariate B-spline basis given by an open knot vector $\Xi := (\xi_1, \ldots, \xi_{n+p+1})$ and degree p. Assume a second knot vector Ξ' which differs from Ξ only in a single knot $\xi' \in [\xi_j, \xi_{j+1})$ for $1 \leq j < n+p+1$, i.e. $\Xi' = (\xi_1, \ldots, \xi_j, \xi', \xi_{j+1}, \ldots, \xi_{n+p+1})$. Then, $\mathcal{S}(\Xi, p) \subset \mathcal{S}(\Xi', p)$.*

Proof. A B-spline basis $N_{i,p} \in \mathcal{B}(\Xi, p)$ can be expressed in $\mathcal{B}(\Xi', p)$ by

$$N_{i,p} = \lambda_i M_{i,p} + (1 - \lambda_{i+1}) M_{i+1,p},\ i = 1, \dots, n,\ M_{i,p} \in \mathcal{B}(\Xi', p)\,, \tag{4.14}$$

$$\text{with coefficients } \lambda_i = \begin{cases} 1 & i = 1, \dots, j-p \\ \frac{\xi' - \xi'_i}{\xi'_{i+p+1} - \xi'_i} & i = j-p+1, \dots, j \\ 0 & i = j+1, \dots, n+1 \end{cases}. \tag{4.15}$$

These coefficients are given in [Piegl and Tiller, 1995, Ch. 5.2, 142 f.], the proof therein is referenced to [De Boor, 2001]. □

The proof is constructive and can be repeated to several different knots. In particular, we say that $\mathcal{B}(\Xi, p)$ in Lemma 4.12 is once globally h-refined, if in each nonempty knot span (ξ_j, ξ_{j+1}) a new know is inserted at its center $\frac{\xi_{j+1}+\xi_j}{2}$. An example of an h-refinement of the basis from Figure 4.1 on page 53 is given in Figure 4.2.

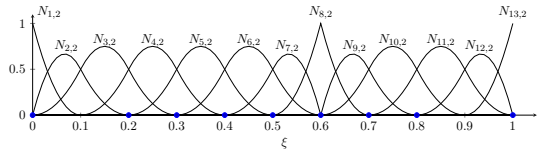


Figure 4.2: Global h-refinement of the basis from Figure 4.1 with the h-refined knot vector $\Xi' = (0, 0, 0, \frac{1}{10}, \frac{1}{5}, \frac{3}{10}, \frac{2}{5}, \frac{3}{5}, \frac{3}{5}, \frac{5}{10}, \frac{7}{10}, \frac{4}{5}, \frac{9}{10}, 1, 1, 1)$

With h-refinement, the number of basis functions is increased by the number of new knots but the degree of the B-splines/NURBS as well as the global continuity stays the same. In a global h-refinement step also the length of the knot spans is decreased by half. In contrast, we introduce now a refinement method where the unique knot spans do not change and the B-spline degree is raised.

Order elevation

We elevate the order or degree of the B-spline basis such that the global continuity is preserved. The latter is important to model a geometry from a lower order space also in the raised order space and preserve its class of continuity.

Lemma 4.13. *Again, let $\Xi := (\xi_1, \dots, \xi_{n+p+1})$ be an open knot vector with degree p and v_j its unique knots collected in Υ. Let μ_j denote the multiplicity of v_j in Ξ. Then, for a knot vector Ξ', with same Υ, which has knot multiplicity $\mu_j + \ell$ for all unique knots v_j in Υ, the spline spaces $\mathcal{S}(\Xi, p) \subset \mathcal{S}(\Xi', p + \ell)$ are nested.*

Proof. For $\ell = 0$ the statement is trivial. Set $\ell = 1$, then a B-spline basis $N_{i,p} \in \mathcal{B}(\Xi, p)$ can be expressed in $\mathcal{B}(\Xi', p + 1)$, [Piegl and Tiller, 1995, Ch. 5.5, p. 188 ff.]. □

The result of Lemma 4.13 is a spline space with raised degree which yields a higher polynomial degree within each element. However, at knots the basis still has the continuity of the original basis which is illustrated in Figure 4.3. We call this process of raising the degree *order elevation* or *p-refinement*.

Remark 4.14. Knot insertion and order elevation works also for multivariate B-splines and NURBS in a canonical way due to the tensor product (like) structure.

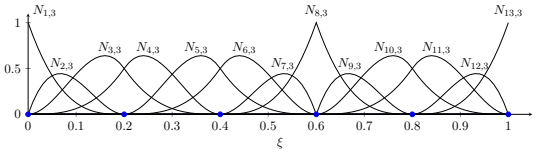


Figure 4.3: A p-refinement of the basis from Figure 4.1 with the new knot vector $\Xi' = (0, 0, 0, 0, \frac{1}{5}, \frac{1}{5}, \frac{2}{5}, \frac{2}{5}, \frac{3}{5}, \frac{3}{5}, \frac{3}{5}, \frac{4}{5}, \frac{4}{5}, 1, 1, 1, 1)$

Both refinement methods lead to nested spline spaces. This is of course important, because we want to be able to represent a given coarse geometry also in the refined spaces, for instance before applying local changes. However, if we want to compare the influence of the global smoothness of the basis functions on the analysis result we cannot use order elevation because it maintains the global regularity of the coarse space by increasing the knot multiplicities. We therefore construct next a space with higher polynomial degree but which has the same unique knots for the sake of practical implementation.

Compatible spaces

We later frequently have computations involving two different spline spaces, for instance having mixed finite element formulations or when we integrate over NURBS geometries with NURBS from a different spline space as integrand. For that, it is computationally efficient and practical if the support of NURBS basis functions from different spaces can be expressed exploiting the tensor product (like) structure. We therefore introduce in this thesis the following characterization of meshes:

Definition 4.15. We call two univariate NURBS spaces $\mathcal{S} := \mathcal{S}(W, \Xi, p)$ and $\mathcal{S}' := \mathcal{S}(W', \Xi', p')$ *compatible* if one set of unique knots, Definition 4.9, contains the other. That is,

$$\Upsilon \subseteq \Upsilon' \text{ or } \Upsilon \supseteq \Upsilon' . \tag{4.16}$$

This can be extended to multivariate spaces $\mathcal{S} := \mathcal{S}(W_1, \Xi_1, p_1; \ldots; W_d, \Xi_d, p_d)$ and $\mathcal{S}' := \mathcal{S}(W_1', \Xi_1', p_1'; \ldots; W_d', \Xi_d', p_d')$ if Υ_i and Υ_i' satisfy (4.16) for each $i = 1, \ldots, d$.

In Figure 4.4 we show an example of an arbitrary mesh compatible to the one from Figure 4.1. The h- and p-refinement discussed before both yield compatible meshes.

Order elevation in Lemma 4.13 preserves the continuity of the spline basis at the knots. Sometimes, however, we want to have spline spaces with a higher order and maximal global continuity, for instance to observe the influence of continuity on solutions of PDEs or in shape optimization. Therefore, we increase the order of a space in a way that yields a compatible mesh and has increased global continuity.

Lemma 4.16. *We enhance an open knot vector* $\Xi := (\xi_1, \ldots, \xi_{n+p+1})$ *with degree* p *by repeating the first and the last knot* ℓ *times, that is,* $\Xi' := (\underbrace{0, \ldots, 0}_{\ell}, \Xi, \underbrace{1, \ldots, 1}_{\ell})$. *Then,* $\mathcal{S}(\Xi, p)$ *is compatible to* $\mathcal{S}(\Xi', p + \ell)$.

Proof. By construction $\Upsilon = \Upsilon'$. □

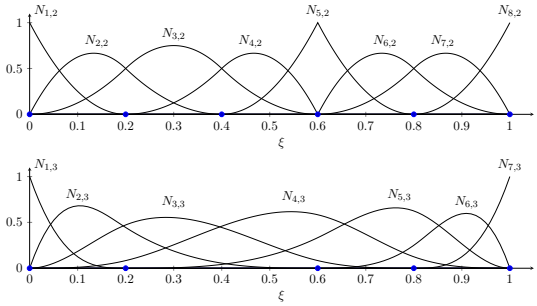


Figure 4.4: Two compatible spline spaces, the basis on the top is the same as in Figure 4.1, the basis on the bottom results from $\Xi = (0, 0, 0, 0, \frac{1}{5}, \frac{3}{5}, \frac{4}{5}, 1, 1, 1, 1)$ and $p = 3$.

In Figure 4.5, enhancing the basis of Figure 4.1 at front and end leads to a basis with higher continuity across knots than using order elevation in Figure 4.3. However, the meshes are compatible and the size of the knot spans does not change.

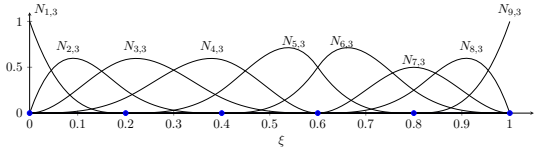


Figure 4.5: Enhancing the knot vector from Figure 4.1 by a 0 at the front and a 1 at the end leads to $\Xi' = (0, 0, 0, 0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{3}{5}, \frac{4}{5}, 1, 1, 1, 1)$ and a compatible mesh.

Next, we characterize a local support relation of NURBS basis functions on compatible spaces.

Lemma 4.17. *Let $\mathcal{S}$, $\mathcal{S}'$ with basis $\mathcal{B} := \mathcal{B}(W, \Xi, p)$, $\mathcal{B}' := \mathcal{B}(W', \Xi', p')$ be two compatible spaces with $\Upsilon \cap \Upsilon' = \Upsilon$. On each element $Q'_j = (v'_j, v'_{j+1})$ with unique knots $v'_j \in \Upsilon'$ live $p' + 1$ basis functions from $\mathcal{B}'$ and $p + 1$ basis functions from $\mathcal{B}$.*

Proof. Denote an element $Q'_j = (v'_j, v'_{j+1})$ with unique knots $v'_j \in \Upsilon'$ and $Q_j = (v_j, v_{j+1})$ with unique knots $v_j \in \Upsilon$. From property (iii) in Lemma 4.10 follows that on each element Q'_j live $p' + 1$ basis functions from $\mathcal{B}'$ and on each Q_j live $p + 1$ from $\mathcal{B}$. We show that there is $Q_k = (v_k, v_{k+1})$ such that $Q'_j \subset Q_k$ which proves the lemma: Since Ξ, Ξ' are both open $0 = v_1 \leq v'_j$. Hence, there is $v_k \in \Upsilon$ with $v_k \leq v'_j$. Choose $v_k = \max\{v \in \Upsilon \colon v \leq v'_j\}$. Since $v_k \leq v'_j < v'_{j+1} \leq 1$, there exists a unique knot $v_{k+1} \in \Upsilon$ such that $v_k < v_{k+1} \leq 1$. From $\Upsilon \cap \Upsilon' = \Upsilon$ we deduce $v'_{j+1} \leq v_{k+1}$. □

We use the definitions from this section to model geometries in the following. Afterwards, we apply them as test functions for the numerical solution of PDEs. Finally, they serve also as variables for shape optimization.

4.2 Geometries in Isogeometric Analysis

In IGA and CAGD, a domain $\Omega \subset \mathbb{R}^d$ can be represented by a geometry map on the unit hypercube $\hat{\Omega} = (0,1)^{\hat{d}}$ which is a linear combination of n NURBS or B-splines

$$G\colon \hat{\Omega} \to \Omega, \quad \hat{x} \mapsto \sum_{i=1}^{n} \sum_{k=1}^{d} X_{i,k} N_i(\hat{x}) e_k \,, \tag{4.17}$$

where we have the B-splines or NURBS $N_i\colon \hat{\Omega} \to \mathbb{R}$ and *control points* $X_i := (X_{i,k})_{k=1,\dots,d}$ in $\mathbb{R}^d$ with standard unit vectors e_k of $\mathbb{R}^d$.

Remark 4.18. Usually in our applications, $\hat{\Omega} = (0,1)^d$, i.e. $\hat{d} = d$. However, this is not mandatory, as for instance shells typically are 2-dimensional manifolds in $\mathbb{R}^3$ with $\hat{d} = 2 < 3 = d$.

More specifically, the geometry map G is generated as follows.

4.2.1 Domain representations

To obtain domains suitable for analysis in IGA, we restrict the domains to a subspace of $\mathcal{S}^d$ containing all $\mathcal{C}^1$-diffeomorphisms with bounded derivatives, where $\mathcal{S}^d := \operatorname{span}\{N_i e_k\colon N_i \in \mathcal{B},\, k = 1,\dots,d\}$ is the direct sum of a tensor product (like) NURBS space $\mathcal{S}$ of dimension n over basis $\mathcal{B}$. Alternatively, [Beirão da Veiga et al., 2014, Assumption 3.1] restrict the geometries to B-spline or NURBS representations that satisfy the Lipschitz conditions (2.32). We have abused the notation and should write R_i when using NURBS. In the following, we denote by $\alpha = (i,k)$ an index tuple such that $N_\alpha := N_i e_k$ and $X_\alpha := X_{i,k}$.

Remark 4.19. $\mathcal{S}^d$ is isomorphic to $\mathbb{R}^{nd}$, hence one can also represent a geometry by the vector of coefficients of the linear combination $G = \mathbf{X}^\intercal \mathbf{N}(\hat{x})$ with $\mathbf{X} := (X_\alpha)_\alpha \in \mathbb{R}^{nd}$, $\mathbf{N} = (N_\alpha)_\alpha$.

Assumption on boundary

To enforce boundary conditions of PDEs with isogeometric analysis, or to move a boundary segment for shape optimization we want to loop over the basis functions with support on the parametric boundary. Therefore, the boundary of the physical domain needs to be a parameterization over the parametric boundary.

Let the boundary of a domain Ω be given by $\partial\Omega = \Gamma \cup \Gamma_N \cup \Gamma_D$, where Γ signifies the moving boundary, Γ_N a boundary with Neumann and Γ_D one with Dirichlet conditions. In the following, we assume that the moving boundary Γ has a preimage

$$\hat{\Gamma} = \{\hat{x} \in \partial\hat{\Omega}\colon\ G(\hat{x}) \in \Gamma\}. \tag{4.18}$$

Furthermore, we require for $\hat{\Gamma}$ that either $(0,1)^{d-1} \times \{\hat{s}\} \subset \hat{\Gamma}$ or $(0,1)^{d-1} \times \{\hat{s}\} \cap \hat{\Gamma} = \emptyset$ for $\hat{s} = 0$ or 1. The same must apply to Γ_D and Γ_N. This means that for each boundary segment Γ, Γ_N and Γ_D, there is a B-spline or NURBS parameterization over a union of sides of the parameter domain in $\mathcal{S}^{d-1}$. We illustrate this for a 2-dimensional example in Figure 4.6. This setting makes sure that an "interesting" boundary segment like a Dirichlet boundary or a moving boundary can always be represented by a union of full faces in the parameter boundary.

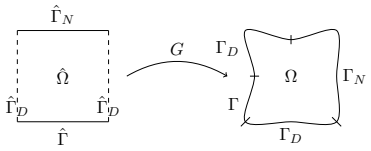


Figure 4.6: We assume that all boundary segments Γ, Γ_N and Γ_D can be mapped by a geometry function G as images of the union of codimension-1 manifolds of the form $[0,1]^{d-1} \times \{0\}$ and $[0,1]^{d-1} \times \{1\}$.

Homogeneous coordinates

The above statements are valid for both, B-splines and NURBS. Note however, that this is only so because the weights are assumed to be fixed. In shape optimization later on in Section 4.4.1 the weights as well as the control points can vary. Furthermore, for the sensitivity analysis we also need the parameterization space $\mathcal{S}^d$ to be a linear space. For that we employ an alternative representation of NURBS geometries using homogeneous coordinates and a perspective map, see [Piegl and Tiller, 1995].

Definition 4.20. A *homogeneous coordinate* vector $\tilde{X}$ in $\mathbb{R}^{d+1}$, with $\tilde{X} := (X^w, X_{d+1})$ and $X^w \in \mathbb{R}^d$, is projected to $\mathbb{R}^d$ by the *perspective map*

$$\mathrm{H}(\tilde{X}) = \begin{cases} X^w / X_{d+1}, & X_{d+1} \neq 0\,, \\ X^w / |X^w|, & \text{else}\,. \end{cases} \tag{4.19}$$

We observe that the weight function $w := \sum_{i=1}^n N_{i,p}\omega_i$ from (4.7) is a positive combination of B-splines

$$w \in \mathcal{S}_+(\Xi, p) := \operatorname{pos} \mathcal{B}(\Xi, p) \subset \mathcal{S}(\Xi, p), \tag{4.20}$$

and a set of NURBS with free weights is given by

$$\mathcal{N}(\Xi, p) := \mathcal{S}(\Xi, p) \times \mathcal{S}_+(\Xi, p). \tag{4.21}$$

Unfortunately, it is not directly obvious how $\mathcal{N}(\Xi, p)$ turns into a vector space because any free weight NURBS function $(s, w) := \frac{s}{w}$ in $\mathcal{N}(\Xi, p)$ is nonlinear w.r.t. the weight functions w, $(s, w_1) + (s, w_2) \neq \frac{s}{w_1 + w_2}$. This also reflects that the rational functions $\frac{1}{x-a}$ and $\frac{1}{x-b}$ are linearly independent for $a \neq b$. In order to restore linearity we resort to homogeneous coordinates and a perspective map.

Free weight NURBS space Given a B-spline space $\mathcal{S}^d$ as in Section 4.2, a rational space $\mathcal{N}$ of free weights is obtained by the direct sum $\mathcal{S}^d \oplus \mathcal{S}$ and the perspective map H in (4.19). More specifically, the vector valued function $(G^w, w) \in \mathcal{S}^{d+1}$, with $G^w \in \mathcal{S}^d$, is mapped from $\mathcal{S}^{d+1} \to \mathcal{S}^d$, which yields a rational d-manifold

$$(G^w, w) \mapsto \begin{cases} G := G^w / w & \text{if } w \neq 0\,, \\ G := G^w / |G^w| & \text{if } w = 0\,. \end{cases} \tag{4.22}$$

By definition of the weight function, the case where $w = 0$ cannot happen in IGA. Finally, $\mathcal{S}^d \oplus \mathcal{S}$ has a basis $\{N_i e_k \colon N_i \in \mathcal{S},\, i = 1, \ldots, n,\, k = 1, \ldots, d+1\}$ and is a linear space which is isomorphic to $\mathbb{R}^{nd} \times \mathbb{R}^d$.

Definition 4.21. Then, the desired *NURBS space with free weights* $\mathcal{N}$ is obtained as

$$\mathcal{N} = \{\mathrm{H}(\tilde{G}) \colon \tilde{G} = (G^w, w) \in \mathcal{S}^d \oplus \mathcal{S}\}\,. \tag{4.23}$$

Applying above definition constructs a rational representation from a given geometry in $\mathcal{N}$. Conversely, we obtain a homogeneous representation from a given rational geometry in the next example.

Example 4.1. Let G be a rational NURBS geometry with multivariate NURBS basis functions from Definition 4.6,

$$R_i(\hat{x}) := \frac{\prod_{j=1}^{d} \omega_{i_j}^{(j)} N_{i_j,p_j}^{(j)}(\hat{x}_j)}{w(\hat{x})} \,. \tag{4.24}$$

We abbreviate $\omega_i := \prod_{j=1}^{d} \omega_{i_j}^{(j)}$ and $N_i := \prod_{j=1}^{d} N_{i_j,p_j}^{(j)}(\hat{x}_j)$ then

$$G = \sum_{i=1}^{n}\sum_{k=1}^{d} X_{i,k} R_i(\hat{x}) e_k = \sum_{i=1}^{n}\sum_{k=1}^{d} X_{i,k}\omega_i \frac{N_i(\hat{x})}{w(\hat{x})} e_k \,. \tag{4.25}$$

This geometry has a representation $\tilde{G} := (G^w, w) \in \mathcal{N}$ with weight function w, modified control points $X_{i,k}^w := X_{i,k}\omega_i$, and

$$G^w := \sum_{i=1}^{n}\sum_{k=1}^{d} X_{i,k}^w N_i(\hat{x}) e_k \,. \tag{4.26}$$

Applying the perspective map H gives back $G = \mathrm{H}(\tilde{G})$.

Control polygon

The control points $X_i \in \mathbb{R}^d$ are a linear approximation to the geometric object. We follow [Beirão da Veiga et al., 2014]:

Definition 4.22. To each B-spline basis function $N_{i,p} \in \mathcal{B}(\Xi, p)$ with knot vector $\Xi = (\xi_1, \ldots, \xi_{n+p+1})$ we associate a *Greville point*

$$\gamma_i := \frac{\xi_{i+1} + \cdots \xi_{i+p}}{p} \text{ for } i = 1, \ldots, n \,. \tag{4.27}$$

These points are the coefficients of identity $\hat{x} = \sum_{i=1}^{n} \gamma_i N_{i,p}(\hat{x})$. For d-variate B-splines the Greville points are calculated for each direction and collected in $\gamma_i := (\gamma_{i_1}, \ldots, \gamma_{i_d})$.

Furthermore, denote by φ_i the standard linear *hat functions* at Greville points γ_j

$$\varphi_i(\gamma_j) := \delta_{i,j} \tag{4.28}$$

with Kronecker delta $\delta_{i,j}$. Then, the *control polygon* is defined as the piecewise linear interpolation

$$C_X(\hat{x}) := \sum_{i=1}^{n} X_i \varphi_i(\hat{x}) \tag{4.29}$$

and under knot insertion and degree elevation the control polygon converges towards the geometry. An important property of NURBS geometries is the *strong convex hull property*. For positive weights ω the NURBS geometry $\Omega = G(\hat{\Omega})$ is contained in the convex hull of its control points, [Piegl and Tiller, 1995, Property P4.25]. This is illustrated in Figure 4.7. The strong property states that $x \in \Omega$ lies in the convex hull of local control points.

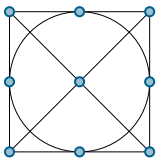


Figure 4.7: A disk representation with NURBS of degree $p = 2$ and its corresponding control points according to the data in Table C.3

Multipatch geometries

Several geometries $\Omega^i \in \mathbb{R}^d$ can be fit together to build one domain Ω. Each part of such a *multipatch geometry* is defined separately over unit hypercubes $(0,1)^{\hat{d}}$. Then all these parts or patches are "glued" together at their mutual faces. That way, complicated geometries can be designed. Say we have n patches, then each patch is described by a geometry function $\Omega^i = G^i(\hat{\Omega}^i)$ and $\bar{\Omega} = \bigcup_{i=1}^n \bar{\Omega}^i$. We now define the global geometry function G piecewise as

$$\Omega = G\left(\bigcup_{i=1}^{n} \hat{\Omega}^i\right) \text{ and } \Omega^i = G|_{\hat{\Omega}^i} = G^i.$$

The glue between patches is the condition that the interface between two patches $I^{i,j} := \bar{\Omega}^i \cap \bar{\Omega}^j$ is an image of the geometry function $I^{i,j} = G(\hat{I}^{ij})$ with $\hat{I}^{ij} := \bar{\hat{\Omega}}^i \cap \bar{\hat{\Omega}}^j$. However, we do not consider multipatches for shape optimization in this work.

4.2.2 Regularity of geometries

The continuity of a spline space $\mathcal{S}$ determines the smoothness of geometry parameterizations. Specifically, the geometry function is k-times continuously differentiable, $G \in \mathcal{C}^k(\hat{\Omega}, \mathbb{R}^d)$ where $k \geq \min(p_i) - 1$ in each component $i = 1, \ldots, d$. However, from Lemma 4.10 we knot that multiple knots diminish the continuity. But also coinciding control points can reduce the continuity by introducing "visual discontinuities" as is shown in properties P3.11 and P3.25 in [Piegl and Tiller, 1995]. Thus, not all important NURBS domains satisfy the conditions in (2.31), they are then for instance not globally one-to-one, which happens e.g. when mapping to triangles or disks.

Example 4.2 (Triangular domain)**.** From the geometry data in Appendix C, Table C.2 we obtain the geometry function

$$G(\hat{x}) = \begin{pmatrix} \hat{x}_1 - \hat{x}_1\hat{x}_2 + \hat{x}_2 \\ \hat{x}_2 \end{pmatrix}$$

of degree 1 which maps the unit square to a right triangle. Clearly, $\det \mathrm{J}_G = 0$ for $(\hat{x}_1, 1)$ and all $\hat{x}_1 \in [0,1]$, which is due to multiple appearing control points.

Example 4.3. The disk in Figure 4.7, page 63, has four singularities at the interpolatory control points, for this and different, though also singular, representations see [Cohen et al., 2010].

Treatment of such parameterizations is on-going work e.g. [Takacs and Jüttler, 2012]. Numerical experiments, however, show that the assumptions may be slackened, as for instance in the case of Example 5.1 of our applications. For our purposes we assume that G fulfills our requirements and ignore a discrepancy in some practical examples.

4.3 Galerkin Projection in Isogeometric Analysis

In this section we describe how to obtain solutions to the PDE problems from Section 2.2 with isogeometric analysis. Basically, IGA uses a change of variables to transform the PDE to a fixed parameter domain paired with an isoparametric Galerkin ansatz. A key feature is that both, the transformation and the test function space, are composed of the same B-splines or NURBS.

In the first part we briefly state main concepts from FEM to be able to compare it to IGA in the second part.

4.3.1 Finite element methods

To solve the PDEs from Section 2.2 numerically, *Galerkin methods* start from their variational form (2.24)

$$u \in \mathcal{V}\colon a(u,v) = l(v) \quad \forall v \in \mathcal{V} \tag{4.30}$$

other than for instance some *collocation methods*, [Quarteroni and Valli, 2008], which use the strong form. Typically for linear elliptic second order PDEs, the test function space is $\mathcal{V} = H^1(\Omega)^m$ over a domain $\Omega \subset \mathbb{R}^d$, where m is the dimension of the solution u. *Conforming* finite element methods take a subspace which has a finite-dimensional basis

$$\mathcal{V}_h := \operatorname{span}\{\varphi_i\colon i = 1,\dots,n\} \subset \mathcal{V}$$

to approximate the solution $u_h \approx u$, where n is the dimension of $\mathcal{V}_h$. Since $\mathcal{V}_h \subset \mathcal{V}$, equation (4.30) is valid for all test functions $v_h \in \mathcal{V}_h$. Any function in $\mathcal{V}_h$ has a basis representation and in particular $u_h = \sum_{i=1}^{n} \mathbf{u}_i \varphi_i$ with coefficients $\mathbf{u}_i \in \mathbb{R}$. Therefore, the finite-dimensional variational equation is equivalent to a linear system

$$u_h \in \mathcal{V}_h\colon a(u_h, v_h) = l(v_h)\, \forall v_h \in \mathcal{V}_h \quad \Leftrightarrow \quad \mathbf{A}\mathbf{u} = \mathbf{F} \tag{4.31}$$

where $q = (\mathbf{u}_1,\dots,\mathbf{u}_n)^\mathsf{T}$ is the vector of solution coefficients, $\mathbf{A}$ is the *stiffness* or *system matrix* with entries $\mathbf{A}_{i,j} = a(\varphi_i, \varphi_j)$ for all $i,j = 1,\dots,n$ and $\mathbf{F}_i = l(\varphi_i)$ is the *load vector* or *right hand side*. The choice of the projection space $\mathcal{V}_h$ determines the finite element method where we use in this thesis only conforming spaces $\mathcal{V}_h \subset \mathcal{V}$. Moreover, the solution u_h is always an element of the test function space $\mathcal{V}_h$ and in our examples $a(\cdot,\cdot)$ is symmetric and positive definite which makes (4.31) a conforming *Ritz-Galerkin discretization*. A recent overview on FEM can be found in [Nochetto et al., 2009]; a classic is [Ciarlet, 1979].

Existence and uniqueness Since we use a conforming Galerkin ansatz, the assumptions for the Lax-Milgram Lemma, Theorem 2.12, are also fulfilled for (4.31) which directly implicates the existence of a unique solution in $\mathcal{V}_h$. For a more general result see also [Nochetto et al., 2009, Thm. 4].

Best approximation property The quality of solutions to the discretized problem is determined by the best approximation property given by Céa's Lemma for coercive bilinear forms. That is, among all functions of the approximation space $\mathcal{V}_h$ it is the solution to (4.30) that comes closest to the real solution. Hence, the error $u - u_h$ is determined by the approximation power of $\mathcal{V}_h$.

Theorem 4.23 (Céa's Lemma).
Let the assumptions in Theorem 2.12 be satisfied with continuity constant M and coercivity constant α from Definition 2.24. Furthermore, let $u \in \mathcal{V}$ denote the solution of (4.30) and $u_h \in \mathcal{V}_h$ its finite element approximation in (4.31). Then we have that

$$\|u - u_h\|_{\mathcal{V}} \leq \frac{M}{\alpha} \min_{v_h \in \mathcal{V}_h} \|u - v_h\|_{\mathcal{V}} .$$

Proof. See [Brenner and Scott, 2008] and the references therein. □

Construction of finite element spaces

The approximation space $\mathcal{V}_h$ should be chosen such that costs of evaluating terms in the variational form (4.31) and the solution of the linear equation system are little, [Nochetto et al., 2009, p. 31]. Therefore, a basis of locally supported functions is attractive because then it is clear that $a(\varphi_i, \varphi_j) = 0$ for most $i \neq j$, and the linear system is sparse. This is achieved by partitioning Ω into small subsets—the elements or cells K—and by building $\mathcal{V}_h$ from the union of local function spaces over these elements. Typically, the elements are simplices like triangles or quadrilaterals in 2 dimensions, and tetrahedrons or hexahedrons in 3 dimensions. "Bent" elements, can be found in isoparametric FEM. The following excerpts [Brenner and Scott, 2008, Ciarlet, 1979] and we refer to them for a more comprehensive introduction to FEM. For the moment, fix the dimension of the solution to $m = 1$ for sake of better readability.

Subdivision of the domain into elements We call a nonempty open domain $K \subset \Omega$ an *element* if it has a piecewise smooth boundary. A finite set $\mathcal{K}_h$ of elements $K \subset \Omega$ is an admissible *mesh* or *triangulation* of Ω if

- it covers Ω completely, $\bar{\Omega} = \bigcup_{K \in \mathcal{K}_h} \bar{K}$,
- two distinct elements K, $K' \in \mathcal{K}_h$ do not overlap, $K \cap K' = \emptyset$,
- there are no hanging nodes, and adjacent elements K, $K' \in \mathcal{K}_h$ share either a vertex or face: if $\bar{K} \cap \bar{K'} \neq \emptyset \Rightarrow \bar{K} \cap \bar{K'} = \{x\}$ or $\bar{K} \cap \bar{K'} = E$ where x is a vertex and E an edge or face of K and K'.

In case of quadrilateral or triangular cells, we need a polygonal domain Ω to fulfill the first point. If the domain is curved we use a polygonal approximation of it instead. We point to Figure 4.8 for an example of a triangulation approximating a curved domain.

Local nodal basis functions On each element $K \in \mathcal{K}_h$, we define a suit of basis functions φ_i for $i = 1, \ldots, dim$ which form a local function space $P(K) = \text{span}\{\varphi_i \colon i = 1, \ldots, dim\}$. These shall be polynomial spaces and each $\pi \in P(K)$ is determined by nodal values: There is a set of linear functionals $\Psi = \{\psi_i \colon P(K) \to \mathbb{R},\ i = 1, \ldots, dim\}$, such that each basis φ_j is uniquely determined by appointing $\psi_i(\varphi_j) = \delta_{ij}$. This justifies in hindsight the hat function notation for the basis elements. The triple $(K, P(K), \Psi)$ is called a *finite element*. We give some examples from [Brenner and Scott, 2008].

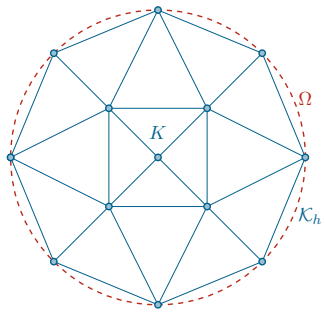
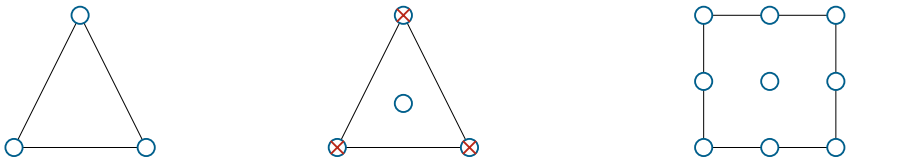


Figure 4.8: Example of a standard FEM triangulation of a disk

Example 4.4. Some common examples of nodal basis functions in $\mathbb{R}^2$ are the following finite elements corresponding to Figure 4.9, which should be understood this way: At nodes × evaluate the gradient and at ○ evaluate the function.

- Let K denote a triangle with vertices z_1, z_2 and z_3 and the local function space be $P(K) = \mathcal{P}_1(K)$ from (4.1), see Figure 4.9a for the position of the nodes. Then $\psi_i(\phi) := \phi(z_i)$ determines the local basis $\{\varphi_1, \varphi_2, \varphi_3\}$ uniquely.
- Let K denote a triangle with vertices z_1, z_2, z_3. The node z_4 is at the barycenter and the local function space be $P(K) = \mathcal{P}_3(K)$. Then the nodal values as given in Figure 4.9b determine the local basis.
- Let K denote a quadrilateral with vertices and midpoints z_i, $i = 1, \dots, 9$ as shown in Figure 4.9c and $P(K) = \mathcal{P}_{Q,2}(K)$ from (4.2). Again consider $\psi_i(\phi) := \phi(z_i)$ which determines the basis $\{\varphi_i,\, i = 1, \dots 9\}$.



(a) Linear Lagrange **(b)** Cubic Hermite **(c)** Bi-quadratic Lagrange

Figure 4.9: Finite element examples of triangular and quadrilateral shapes where the marked vertices and midpoints are nodes, see Example 4.4 for the meaning of different colored nodes

Global function space To obtain the approximation space $\mathcal{V}_h$, we stitch together the locally defined finite elements. For a triangulation $\mathcal{K}_h$ with finite elements $(K, P(K), \Psi)$ for each $K \in \mathcal{K}_h$ the finite element space is specified by

$$\mathcal{V}_h = \{v_h \colon \Omega \to \mathbb{R}^m,\, v_h|_K \in P(K)\, \forall K \in \mathcal{K}_h\}\,.$$

To consider Dirichlet boundary conditions we insert in above definition

$$\mathcal{V}_h = \{v_h \colon \Omega \to \mathbb{R}^m,\, v_h|_K \in P(K)\, \forall K \in \mathcal{K}_h \text{ and } v_h|_{\Gamma_D} = 0\}$$

which we write simply as $\mathcal{V}_h = \{v_h \colon \Omega \to \mathbb{R}^m,\, v_h|_K \in P(K)\, \forall K \in \mathcal{K}_h\} \cap \mathcal{V}$. The finite element spaces resulting from Example 4.4 are conforming spaces as shown in

[Brenner and Scott, 2008]. We collect the evaluation points z of all elements in $\mathcal{N}_h$; each such node corresponds to a basis function φ. The global dimension n of $\mathcal{V}_h$ is defined by the number of basis functions.

Definition 4.24. Let h_K denote the diameter of an element K, then the largest element size

$$h := \max_{K \in \mathcal{K}_h} h_K \tag{4.32}$$

is the *discretization parameter* and gives meaning to the index of $\mathcal{V}_h$.

We always assume triangulations where the cells are not "distorted much":

Definition 4.25. A triangulation is *shape-regular* if there exists $c > 0$ such that $\frac{h_K}{\rho_K} \leq c$ for all $K \in \mathcal{K}_h$ where ρ_K denotes the radius of the in-circle in K.

Convergence and error analysis Consider Lagrange elements where the evaluation points $z_1, \ldots, z_{dim}$ are the vertices of an element K. Due to the nodal basis condition $\varphi_i(z_j) = \delta_{ij}$, we can define a *local interpolator*

$$i_h^K \colon \mathcal{C}(K) \to P(K), \quad i_h^K \phi = \sum_{i=1}^{dim} \phi(z_i)\varphi_i \,. \tag{4.33}$$

For all K in $\mathcal{K}_h$ we join them together to obtain a *global interpolator* with $i_h\phi|_K = i_h^K\phi$ and nodes $z_i \in \mathcal{N}_h$:

$$i_h \colon \mathcal{C}(\bar{\Omega}) \to \mathcal{V}_h, \quad i_h\phi = \sum_{i=1}^{n} \phi(z_i)\varphi_i \,. \tag{4.34}$$

For a weak solution $u \in \mathcal{V}$ of (4.30) and its finite element approximation $u_h \in \mathcal{V}_h$ from (4.31), we have a convergence result $\|u - u_h\|_{\mathcal{V}} \to 0$ as h approaches 0. Furthermore, with Céa's Lemma 4.23 the interpolation error of a solution u with finite elements gives an estimate of the approximation quality of the finite dimensional solution u_h. If $u \in H^{k+1}(\Omega)$ the following error estimate holds

$$\|u - u_h\|_{H^1(\Omega)} \leq ch^k |u|_{H^{k+1}(\Omega)} \,. \tag{4.35}$$

With an Aubin-Nitsche trick also an estimate optimal in L^2 can be obtained

$$\|u - u_h\|_{L^2(\Omega)} \leq ch^{k+1} |u|_{H^{k+1}(\Omega)} \,. \tag{4.36}$$

The two results are for instance shown as Theorem 10.4.1 and equation (10.4.11) in [Atkinson and Han, 2001] where also the domain smoothness plays an important role for u having higher regularity.

Remark 4.26. In particular, the proofs are based on the nodal interpolation as $\|u - u_h\|_{H^1(\Omega)} \leq c\|u - i_h u\|_{H^1(\Omega)}$ and $\|u - i_h u\|_{H^1(\Omega)} \leq c(\sum_{K \in \mathcal{K}_h} \|u - i_h^K u\|_{H^1(K)})^{1/2}$. Therefore, the element-wise estimates $\|u - i_h^K u\|_{H^1(K)}$ are needed. Usually, they are obtained by looking for this property on a reference element $\hat{K}$ with the Bramble-Hilbert lemma and then scaled to the physical elements.

Reference element Assume there is a reference element $\hat{K}$ with dim nodes $\hat{z}_i$, $i = 1, \dots, dim$, which defines a finite element $(\hat{K}, \hat{P}, \hat{\Psi})$. Furthermore, let $\hat{\psi} \in \hat{\Psi}$ describe Lagrange elements, i.e. evaluate $\hat{\psi}_i(\hat{\pi}) = \hat{\pi}(\hat{z}_i)$ for $\hat{\pi} \in \hat{P}$. Given a one-to-one mapping

$$\sigma \colon \hat{K} \to \mathbb{R}^d \text{ such that each space component } \sigma_i \in \hat{P} \text{ for } 1 \le i \le d\,, \tag{4.37}$$

we get physical elements (K, P, Ψ) by the transformation

$$K = \sigma(\hat{K}), \quad P = \{\pi = \hat{\pi} \circ \sigma^{-1} \colon \hat{\pi} \in \hat{P}\}, \quad \Psi = \{\psi_i(\pi) = \pi(\sigma(\hat{z}_i)),\, \pi \in P\}\,. \tag{4.38}$$

With this rule we kill two birds with one stone. First, for *affine transformations* $\sigma = B\hat{x} + b$ the computation of the variational terms is pulled back to the reference domain $\hat{K}$ which makes the assembly of stiffness matrix and load vector easier to implement and less costly. Then, secondly, one can use the two estimates

$$|\hat{v}|_{W^{k,p}(\hat{K})} \le C\|B\|^r |\det B|^{-\frac{1}{p}} |v|_{W^{r,p}(K)} \quad \forall v \in W^{r,p}(K) \tag{4.39}$$

$$|v|_{W^{r,p}(K)} \le C\|B^{-1}\|^r |\det B|^{\frac{1}{p}} |\hat{v}|_{W^{k,p}(\hat{K})} \quad \forall \hat{v} \in W^{r,p}(\hat{K}) \tag{4.40}$$

as scaling argument in the convergence analysis for FEM in Remark 4.26. The inequalities above can be found in [Ciarlet, 1979, Thm. 3.1.2, p. 117] and are consequences of Theorem 2.16 and the formula of change of variables in integrals. Third, if σ is more complicated, it leads to *isoparametric* finite element which allows elements with curved boundaries.

Isoparametric FEM The name "isoparametric" element alludes to the fact that $\hat{P}$ is used for the local function space and also for the construction of the transformation σ in equation (4.37). We do not go into details of constructing such elements and only point towards Chapter 4.4 in [Ciarlet, 1979] where examples can also be found. However, we do quote from there the observation that "an isoparametric element is not directly determined by a mapping" σ but again by the nodes

$$\sigma \colon \hat{x} \in \hat{K} \mapsto \sigma(\hat{x}) = \sum_{i=1}^{dim} \hat{\pi}(\hat{x})\sigma(\hat{z}_i)\,,$$

thus still having a local character. Isoparametric FEM is the namesake of isogeometric analysis where the same principles are applied. Nevertheless, IGA does have a globally determined mapping σ contrary to the quote above and, moreover, is not based on a nodal basis.

4.3.2 Isogeometric analysis

Having established conforming Ritz-Galerkin methods and the isoparametric concept, we are not far from isogeometric analysis, the main difference being the choice of basis functions as B-splines or NURBS. We recall that the starting point is the discretization of the variational equation (2.33)

$$u \in \mathcal{V} \colon a(u, v) = l(v) \quad \forall v \in \mathcal{V}\,, \tag{4.41}$$

where $\mathcal{V} = H_0^1(\Omega)^m$, $\Omega \subset \mathbb{R}^d$, and m the dimension of the solution u. We know from Theorem 2.16 that a change of variables holds for a $\mathcal{C}^1$-diffeomorphism $G\colon \hat{\Omega} \to \Omega$ with bounded derivatives, alternatively satisfying (2.32), and we obtain the equivalent, transported, variational equation (2.34):

$$\hat{u} \in \hat{\mathcal{V}}\colon \hat{a}(G)(\hat{u}, \hat{v}) = \hat{l}(G)(\hat{v}) \quad \forall \hat{v} \in \hat{\mathcal{V}}\,, \tag{4.42}$$

with $\hat{u} = u \circ G$ and $\hat{\mathcal{V}} = H_0^1(\hat{\Omega})^m$. The key to IGA is that the diffeomorphism G is a B-spline or NURBS representation from Section 4.2. We next describe, how the projection space is discretized by using the same basis functions for $\mathcal{V}$ as for the geometry G. By doing so, an analogy to FEM meshes and elements can be established.

Projection space in IGA

The fundamental idea in IGA is that the domain is given by a B-spline or NURBS representation G and uses therefore a finite-dimensional subspace

$$\mathcal{V}_h := \{N \circ G^{-1}\colon N \in \mathcal{S}\}^m \cap \mathcal{V} = \mathcal{S}^m \circ G^{-1} \cap H^1_{\Gamma_D}(\Omega)^m \tag{4.43}$$

for the discrete Galerkin formulation $u_h \in \mathcal{V}_h\colon a(u_h, v_h) = l(v_h) \quad \forall v_h \in \mathcal{V}_h$. Practically, IGA uses equation (4.42) and solves for $\hat{u}_h \in \hat{\mathcal{V}}_h := \mathcal{S}^m \cap \hat{\mathcal{V}}$:

$$\hat{a}(G)(\hat{u}_h, \hat{v}) = \hat{l}(G)(\hat{v}) \quad \forall \hat{v} \in \hat{\mathcal{V}}_h. \tag{4.44}$$

As above, the hat notation $\hat{}$ indicates that a function is defined over the parameter space $\hat{\Omega} = (0,1)^d$ or that a variable is from $\hat{\Omega}$.

Remark 4.27. In fact we could use any G that satisfies (2.31) or (2.32) and are not restricted by theory to use a parameterization from a spline space, let alone the same one as the test functions. But particularly, given two different spline spaces, NURBS or B-splines, $\mathcal{S}_{geo}$ and $\mathcal{S}_{sim}$, we could choose G from $\mathcal{S}^d_{geo}$ and $\hat{\mathcal{V}}_h = \mathcal{S}^m_{sim} \cap H^1_{\Gamma_D}(\hat{\Omega})^m$ and (4.44) still yields a legitimate solution $\hat{u}_h = u_h \circ G^{-1}$, with $\hat{u}_h \in \hat{\mathcal{V}}_h$.

All in all, isogeometric analysis has the following two main ingredients:

- A *geometry function* $G \in \mathcal{S}^d_{geo}$ with tensor product spline space $\mathcal{S}_{geo}$ with basis $\mathcal{B}_{geo}$

$$G\colon \hat{\Omega} \to \Omega, \quad \hat{x} \mapsto \sum_{\substack{\alpha=(i,k),\\ i=1,\dots,n_{geo},\\ k=1,\dots,d}} X_\alpha N_\alpha \tag{4.45}$$

  with $n_{geo} = \dim \mathcal{S}_{geo}$, $N_\alpha = N_i e_k \in \mathbb{R}^d$ and $N_i \in \mathcal{B}_{geo}$ and

- a *Galerkin projection* to the space $\hat{\mathcal{V}}_h = \mathcal{S}^m_{sim}$ with tensor product spline space $\mathcal{S}_{sim}$ with basis $\mathcal{B}_{sim}$, i.e.

$$\hat{u}_h = \sum_{\substack{\alpha=(i,k),\\ i=1,\dots,n_{sim},\\ k=1,\dots,m}} q_\alpha M_\alpha \tag{4.46}$$

  with $n_{sim} = \dim \mathcal{S}_{sim}$, $M_\alpha = N_i e_k \in \mathbb{R}^m$ and $N_i \in \mathcal{B}_{sim}$.

Remark 4.28. Above, substitute N_i by R_i for a notation with rational B-splines.

Mesh and elements

In the underlying parameter domain, the knot vectors form a grid which allows to speak of elements and the geometry function projects them to physical elements.

Isogeometric mesh Let Ξ_i denote knot vectors for $i = 1, \ldots, d$ with knots ξ^i_j. Then, the cross product of the knot vectors $\Xi_1 \times \cdots \times \Xi_d$ forms a grid of elements Q defined by the knot span products

$$Q_{j_1,\ldots,j_d} = (\xi^1_{j_1}, \xi^1_{j_1+1}) \times \cdots \times (\xi^d_{j_d}, \xi^d_{j_d+1}). \tag{4.47}$$

We call the set of all nonempty elements due to Ξ_i, $i = 1, \ldots, d$ the *isogeometric mesh* or *grid*

$$\hat{\mathcal{K}}_h = \{Q_{j_1,\ldots,j_d} \neq \emptyset \colon j_i = 1, \ldots, n_i + p_i \text{ and } i = 1, \ldots, d\}. \tag{4.48}$$

The size of each element is expressed by its diameter h_Q. Again we define a global mesh parameter h similar to (4.32) by

$$h = \max_{Q \in \hat{\mathcal{K}}_h} h_Q \,. \tag{4.49}$$

Since NURBS are defined over several knot spans, let N denote a basis function of the tensor product spline space $\mathcal{S}$ from Definition 4.4, then

$$\hat{\Pi}_h(N) := \{Q \in \hat{\mathcal{K}}_h \colon Q \subset \operatorname{supp}(N)\} \tag{4.50}$$

gives the elements in the support of N. Note, that for h-refined knot vectors Ξ'_i we obtain a mesh $\hat{\mathcal{K}}_{h'}$ with a new mesh parameter $h' \leq h$. Given N on the coarse mesh $\hat{\mathcal{K}}_h$ we can still evaluate $\hat{\Pi}_{h'}(N)$ since $\bigcup\{Q \in \hat{\Pi}_{h'}(N)\} = \bigcup\{Q \in \hat{\Pi}_h(N)\}$. However, for a refined basis function N defined over $\hat{\mathcal{K}}_{h'}$ it makes no sense to search for coarse elements in the support $\hat{\Pi}_h(N)$.

Physical mesh The geometry function maps the parameter elements to a physical "triangulation"

$$\mathcal{K}_h = \{K \subset \Omega \colon K = G(Q) \text{ for } Q \in \hat{\mathcal{K}}_h\} \,. \tag{4.51}$$

As in FEM we favor homogeneous sized elements, therefore we assume shape regularity which implies that $h_K \simeq h_Q$, see also [Beirão da Veiga et al., 2014, Assumption 3.1].

Convergence and error analysis

We proceed as in the convergence analysis for FEM and go back to the interpolant in (4.33). There, we have the following:

- On page 65, a nodal basis is defined from a *dual basis* ψ, $\psi_i(\varphi_j) = \delta_{ij}$ and in particular $\psi_i(\phi) = \phi(z_i)$ for a function ϕ.
- The relation to a reference element is given by a mapping of the reference nodes $z_i = \sigma(\hat{z}_i)$ in (4.38).

From this summary we see that the interpolant in (4.34) has the representation

$$i_h\phi(x) = \sum_{i=1}^{n} \psi_i(\phi)\varphi_i(x) = \sum_{i=1}^{n} \phi(z_i)\varphi_i(x) \,. \tag{4.52}$$

Interpolation on the parameter domain In IGA the basis functions are not interpolatory which means that although we can use a dual basis with $\psi_i(\varphi_j) = \delta_{ij}$ it does not have the nodal property $\psi_i(\phi) = \phi(z_i)$. Instead, [Beirão da Veiga et al., 2014] introduces a multivariate *quasi-interpolation operator* from a suffiently regular function space over the parameter domain to a multivariate B-spline space $\mathcal{S}$ of dimension n, for instance,

$$\hat{\imath}_h \colon L^2(\hat{\Omega}) \to \mathcal{S}, \quad \hat{\imath}_h\hat{\phi}(\hat{x}) = \sum_{i=1}^{n} \Lambda_i(\hat{\phi}) N_i(\hat{x}) \,. \tag{4.53}$$

that satisfies on the parameter domain $\Lambda_i(N_j) = \delta_{ij}$ for B-spline N_j and

$$\Lambda_i(\hat{\phi}) = \int_{\operatorname{supp} N_j} \hat{\phi}(\hat{x}) g_i(\hat{x}) \,\mathrm{d}\hat{x} \,. \tag{4.54}$$

Here, g_i is a functional involving the derivative of a *transition function* from a B-spline basis N_i. Both g_i and Λ_i depend on the univariate spline spaces which build the tensor product (like) space $\mathcal{S}$. Since we do not use the properties of the dual basis Λ_i and neither need its exact construction in the following, we refer the reader to [Beirão da Veiga et al., 2014] and the reference to [Schumaker, 1981, Theorem 4.37] therein. Note, that for functions $\hat{\phi}$ in $H^{k+1}(\hat{\Omega})$ a different operator is needed to quasi-interpolate it with B-splines. It also is expressed with an L^2 projection as above in (4.54) under the assumption that the degree of the B-spline is $p \geq 2k+1$. For our purposes it is enough to know that there are such interpolators in IGA and make no difference in notation for $\hat{\imath}_h \colon H^{k+1}(\hat{\Omega}) \to \mathcal{S}$.

Interpolation on the physical domain From the quasi-interpolator on the parameter domain, we arrive at a projector for functions in X on the physical domain by first transforming it with the geometry function to the parameter domain, interpolate it there and then project it back to the physical domain by the inverse of the geometry function. Let X be a space of functions with sufficient regularity, e.g. $X = L^2(\Omega)$ or $H^{k+1}(\Omega)$ for instance, then

$$i_h \colon X \to \mathcal{V}_h, \quad i_h\phi = \hat{\imath}_h(\phi \circ G) \circ G^{-1} \,. \tag{4.55}$$

Note, that for NURBS we insert the weight function in between

$$i_h \colon X \to \mathcal{V}_h, \quad i_h\phi = \frac{\hat{\imath}_h\big(w(\phi \circ G)\big)}{w} \circ G^{-1} \,. \tag{4.56}$$

In tune with Remark 4.27 this projector also supports a non-isoparametric approach as noted in [Beirão da Veiga et al., 2014, Rem. 3.4]: The geometry function G may be from a different NURBS space than the interpolator $\hat{\imath}_h$ although the authors there state this only for two differently refined spaces.

Error analysis With the interpolators we state now the error analysis as for FEM. We start with the estimates in the parameter spaces and then give those over the physical domain.

Lemma 4.29. *Let $\mathcal{S} := \mathcal{S}(W_1, \Xi_1, p_1; \ldots; W_d, \Xi_d, p_d)$ denote the tensor product (like) spline approximation space for $\hat{u} \in H^s(\hat{\Omega})$. Assume that it has the same polynomial degree in each univariate component $p_i = p$, $i = 1, \ldots, d$ and that the knot multiplicity of each knot in knot vector Ξ_i is μ_i. Then the basis functions have regularity $p - \mu_i$ across the i^{th} space coordinate. Therefore, let $p+1 \geq k \geq d+\sum_{i=1}^{d}(p-\mu_i)$. Furthermore, let $0 \leq s \leq \min_{i=1,\ldots d}\{p - \mu_i\} + 1$ then for an element $Q \in \hat{\mathcal{K}}_h$ the local estimate holds*

$$|\hat{u} - \hat{\imath}_h \hat{u}|_{H^s(Q)} \leq C h_Q^{k-s} |\hat{u}|_{H^k(Q)} \,. \tag{4.57}$$

Proof. The local estimate is given in [Beirão da Veiga et al., 2014], Proposition 4.14 and Corollary 4.15. □

Lemma 4.30. *We have the local estimate for $u \in H^k(K)$, where K is a transformed element in the physical domain given by $G(Q)$ for $Q \in \hat{\mathcal{K}}_h$. There, the isogeometric interpolant i_h maps from $H^s(\Omega)$ to $\mathcal{V}_h$ by means of (4.56), which yields*

$$|u - i_h u|_{H^s(K)} \leq C h^{k-s} \|u\|_{H^k(K)} \tag{4.58}$$

and the same for the whole domain Ω if $u \in H^k(\Omega)$.

Proof. The above lemma is given in [Beirão da Veiga et al., 2014] as Theorems 4.23 and 4.24. □

Remark 4.31. Note, that for $s = 0$ we obtain the same estimate as in the L^2 norm in (4.36) for FEM. An estimate for B-splines of maximum smoothness is discussed in the preprint [Takacs and Takacs, 2015].

4.4 Shape Calculus in Isogeometric Analysis

With the shape optimization theory from Chapter 3 and the B-spline and NURBS spaces from this chapter we have now the tools at hand to formulate isogeometric discretizations for shape optimization. We recall that to solve a shape optimization problem (1.1) numerically, we have two options: We can either *(i)* use the necessary optimality conditions for the infinite-dimensional problem in Section 3.4 and then discretize both controls and state. Or *(ii)* we first discretize both control and state variables in the SOP (1.1), and then formulate optimality conditions for the resulting finite-dimensional problem. Either way we then have a nonlinear finite-dimensional optimization problem for which we use numerical methods like a gradient descent method to find an optimal shape.

Basically in a gradient descent method a sequence of domains $\{\Omega_k\}_{k=0,1,\ldots}$ is generated by an update rule

$$\Omega_{k+1} = (id + t s_k)(\Omega_k) \tag{4.59}$$

where s_k is a descent direction, typically $s_k = -\nabla_{\Omega_k} J(\Omega_k)$, and t a step size. The update rule of equation (4.59) employs a descent direction resp. gradient from a real vector space for scheme *(ii)* while in case *(i)* the descent direction is an element of a function space. In [Haslinger and Mäkinen, 2003], the *discretize first* point of view is

described for FEM, whereas in [Delfour and Zolésio, 2011] the *optimize first* approach in the form of Lagrange multipliers can be found, for instance. Chapter 3 discussed the use of these approaches in IGA, namely,

- [Wall et al., 2008], [Qian, 2010], [Nguyen, 2012], [Kiendl et al., 2014] for the *discretize first*, and
- [Blanchard et al., 2013], [Bandara et al., 2015] for the *optimize first* ansatz.

Of special interest is [Qian, 2010] where sensitivities for a structural design problem with IGA in the *discretize first* setting are developed also w.r.t. NURBS weights. As we focus more on the *optimize first* scheme, an important result of this chapter is a formulation for weight optimization of NURBS geometries in this setting, and a comparison of *discretize first* and *optimize first* for IGA.

In the following, we formulate the *optimize first–discretize then* scheme *(i)* for IGA which uses the abstract framework from Section 3.4. In particular, we derive the sensitivities w.r.t. NURBS weights also in this framework. Thereafter, the contrary case *(ii) discretize first–optimize then* is treated. Subsequently, we show that both methods eventually lead to the same discrete system for our class of problems.

4.4.1 Optimize first, discretize then

In this case, we discretize the Karush-Kuhn-Tucker system from Section 3.4. Recall that we couple cost functional and state equation in the Lagrange function

$$\mathcal{L}(\Omega, u, v) := J(u, \Omega) + l(v) - a(u, v), \tag{4.60}$$

or in the isogeometric version with a parameterization $G_t = G + t\hat{q}$,

$$\mathcal{L}(G_t, \hat{u}_t, \hat{v}) := \hat{J}(G_t, \hat{u}_t) + \hat{l}(G_t)(\hat{v}) - \hat{a}(G_t)(\hat{u}, \hat{v}) \tag{4.61}$$

where all variables $\hat{q}$, $\hat{u}$ and $\hat{v}$ are independent. Under the assumption that the state u and thus $\hat{u} = u \circ G$, is shape differentiable, the necessary optimality conditions are given by the KKT system $\nabla\mathcal{L} = 0$, i.e., at an optimum $(\hat{u}, \hat{z}, G)$ it holds

$$\mathrm{d}_{\hat{v}}\mathcal{L}(G, \hat{u}, \hat{z}; \delta\hat{v}) = 0 \quad \forall \delta\hat{v} \in \hat{\mathcal{V}}, \tag{4.62}$$

$$\mathrm{d}_{\hat{u}}\mathcal{L}(G, \hat{u}, \hat{z}; \delta\hat{u}) = 0 \quad \forall \delta\hat{u} \in \hat{\mathcal{V}}, \tag{4.63}$$

$$\mathrm{d}_{\hat{q}}\mathcal{L}(G, \hat{u}, \hat{z}; \delta\hat{q}) = 0 \quad \forall \delta\hat{q} \in \hat{\mathcal{Q}}\,. \tag{4.64}$$

Explicitly, the KKT system in IGA is given from Lemma 3.25 which we restate here.

Lemma 4.32. *The first order necessary optimality system of using* (4.62)–(4.64) *in isogeometric shape optimization is*

$$\hat{a}(G)(\hat{u}, \delta\hat{v}) = \hat{l}(G)(\delta\hat{v}) \qquad \forall \delta\hat{v} \in \hat{\mathcal{V}} \qquad \textit{state equation,} \tag{4.65}$$

$$\mathrm{d}_{\hat{u}}\hat{J}(G, \hat{u}; \delta\hat{u}) = \hat{a}(G)(\delta\hat{u}, \hat{z}) \quad \forall \delta\hat{u} \in \hat{\mathcal{V}} \qquad \textit{adjoint equation,} \tag{4.66}$$

with $\mathrm{d}_{\hat{u}}\hat{J}(G, \hat{u}; \delta\hat{u}) = \mathrm{d}_t\hat{J}(G, \hat{u} + t\delta\hat{u})|_{t=0}$. *The shape gradient is formed by directional derivatives*

$$\mathrm{d}_{\hat{q}}\mathcal{L}(G, \hat{u}, \hat{v}; \delta\hat{q}) = \mathrm{d}_{\hat{q}}\hat{J}(G, \hat{u}; \delta\hat{q}) + \mathrm{d}_{\hat{q}}\hat{l}(\hat{v}; \delta\hat{q}) - \mathrm{d}_{\hat{q}}\hat{a}(\hat{u}, \hat{v}; \delta\hat{q}) \tag{4.67}$$

with $\mathrm{d}_{\hat{q}}\hat{J}(G, \hat{u}; \delta\hat{q})$ *given by expression* (3.53) *in Lemma 3.24,*

$$\mathrm{d}_{\hat{q}}\hat{l}(\hat{v}; \delta\hat{q}) = \mathrm{d}_t\hat{l}(G + t\delta\hat{q})(\hat{v})|_{t=0} \text{ and } \mathrm{d}_{\hat{q}}\hat{a}(G)(\hat{u}, \hat{v}; \delta\hat{q}) = \mathrm{d}_t\hat{a}(G + t\delta\hat{q})(\hat{u}, \hat{v})|_{t=0}.$$

This systems turns fully discrete by projecting $\hat{\mathcal{V}}$ onto $\hat{\mathcal{V}}_h$ and by choosing $\hat{\mathcal{Q}} \subset \mathcal{S}^d$ which means $G \in \mathcal{S}^d$ as well.

Shape sensitivities for optimal weights

In [Qian, 2010], shape optimization simultaneously over control points and weights has been performed for an example of linear elasticity, based on the *discretize first* ansatz. Together with the framework of Section 3.4, we now extend weight optimization to general elliptic PDEs for the *optimize first* scheme. In Section 4.2 we introduced B-spline and NURBS spaces. For the latter we assumed a fixed vector of weights W, see Remark 4.1.3. In shape optimization, however, weights might be another instrument of fine-tuning. But, if W is not fixed, a much larger space than $\mathcal{S}$ of (4.6) is available in order to search for the best shape. Different to the notation of Remark 4.1.3, we have free weights. Therefore we use the alternative representation in Definition 4.21 in the following sensitivity formula.

Lemma 4.33. *For a shape functional J as in (3.31), we obtain the isogeometric shape sensitivities in direction $\tilde{\theta} \in \mathcal{N}$ for $G = \mathrm{H} \circ \tilde{G}$ and $\delta\hat{q} = \mathrm{H} \circ \tilde{\theta}$ as*

$$\begin{aligned}
&\mathrm{d}\hat{J}(G; \delta\hat{q}) \\
&\quad = \int_{\hat{\Omega}} (\nabla j_1 \circ G \cdot \mathrm{DH} \circ \tilde{G}\tilde{\theta} + j_1 \circ G \operatorname{tr}(\mathrm{J}_G^{-1} \dot{D}))|\det \mathrm{J}_G| \,\mathrm{d}\hat{x} && (4.68)\\
&\quad + \int_{\hat{\Gamma}} \Bigg(\nabla j_2 \circ G \cdot \mathrm{DH} \circ \tilde{G}\tilde{\theta} |\mathrm{J}_G^{-\intercal} \hat{n}| + j_2 \circ G |\mathrm{J}_G^{-\intercal} \hat{n}| \operatorname{tr}(\mathrm{J}_G^{-1} \dot{D}) + && (4.69)\\
&\qquad - j_2 \circ G \frac{\mathrm{J}_G^{-\intercal} \dot{D}^{\intercal} \mathrm{J}_G^{-\intercal}}{|\mathrm{J}_G^{-\intercal} \hat{n}|} \Bigg) |\det \mathrm{J}_G| \,\mathrm{d}\hat{x} && (4.70)
\end{aligned}$$

with $\dot{D} := \mathrm{d}_t D(H \circ \tilde{G}_t)|_{t=0}$.

Proof. Since we use homogeneous coordinates for $\tilde{G}$ and the perspective map H, this implies that by the substitution of variables in the shape functional we have

$$\begin{aligned}
\hat{J}(G) = &\int_{\hat{\Omega}} j_1 \circ (\mathrm{H} \circ \tilde{G})| \det(\mathrm{D}(\mathrm{H} \circ \tilde{G}))| \,\mathrm{d}\hat{x} \\
&+ \int_{\hat{\Gamma}} j_2 \circ (\mathrm{H} \circ \tilde{G})| \,\mathrm{D}(\mathrm{H} \circ \tilde{G})^{-1} \hat{n}|| \det(\mathrm{D}(\mathrm{H} \circ \tilde{G}))| \,\mathrm{d}\hat{x}.
\end{aligned}$$

For the limit $\mathrm{d}_t J(G_t)|_{t=0}$ with $G_t = \mathrm{H} \circ (\tilde{G} + t\tilde{\theta})$ we need the transformations

$$\mathrm{J}_G = \mathrm{D}G = \mathrm{D}(\mathrm{H} \circ \tilde{G}) = \mathrm{DH} \circ \tilde{G} \,\mathrm{D}\tilde{G} \text{ and} \tag{4.71}$$

$$\mathrm{DH} \circ \tilde{G} = \frac{1}{w} \begin{pmatrix} I_d, & -G \end{pmatrix}, \tag{4.72}$$

with identity matrix $I_d \in \mathbb{R}^{d \times d}$ and assuming positive weights. Therefore, it holds for $\tilde{G}_t := \tilde{G} + t\tilde{\theta}$

$$\dot{D} = \mathrm{d}_t \,\mathrm{D}(\mathrm{H} \circ \tilde{G}_t)|_{t=0} = \mathrm{D}^2 \mathrm{H} \circ \tilde{G} \odot \tilde{\theta} \,\mathrm{D}\tilde{G} + \mathrm{DH} \circ \tilde{G} \,\mathrm{D}\tilde{\theta}, \tag{4.73}$$

where $\mathrm{D}^2\mathrm{H}(x) \odot v := (\mathrm{d}_{x_1} \mathrm{DH}\, v, \ldots, \mathrm{d}_{x_{d+1}} \mathrm{DH}\, v) \in \mathbb{R}^{d\times d+1}$ for x and $v \in \mathbb{R}^{d+1}$. With this notation we get explicitly

$$\begin{aligned}
&\mathrm{D}^2\mathrm{H} \circ \tilde{G} \odot \tilde{\theta} = \\
&= \frac{1}{w^2}\left((0_d, -e_1)\tilde{\theta}, \ldots, (0_d, -e_d)\tilde{\theta}, (-I_d, 2G)\tilde{\theta}\right) && (4.74)\\
&= \begin{cases} \frac{1}{w^2}\left(0_d, \quad -\delta\hat{q}^w\right) & , \text{if } \tilde{\theta}_{d+1} = 0 \\ \frac{-\tilde{\theta}_{d+1}}{w^2}\left(I_d, \frac{1}{\tilde{\theta}_{d+1}}\delta\hat{q}^w - 2G\right) & , \text{if } \tilde{\theta}_{d+1} \neq 0\,. \end{cases} && (4.75)
\end{aligned}$$

From the implicit function theorem we obtain

$$\begin{aligned}
0 &= \mathrm{d}_t I_d = \mathrm{d}_t\big(\mathrm{D}(\mathrm{H}\circ\tilde{G}_t)^{-1}\,\mathrm{D}(\mathrm{H}\circ\tilde{G}_t)\big)|_{t=0} \\
&= \mathrm{d}_t\big(\mathrm{D}(\mathrm{H}\circ\tilde{G}_t)^{-1}\big)|_{t=0}\,\mathrm{D}(\mathrm{H}\circ\tilde{G}) + \mathrm{D}(\mathrm{H}\circ\tilde{G})^{-1}\dot{D}\,.
\end{aligned}$$

Hence,

$$\mathrm{d}_t\big(\mathrm{D}(\mathrm{H}\circ\tilde{G}_t)^{-1}\big)|_{t=0} = -\,\mathrm{J}_G^{-1}\,\dot{D}\,\mathrm{J}_G^{-1},$$

which completes the proof. □

4.4.2 Discretize first, optimize then

Spline spaces are isomorphic to real vector spaces, which means that $\mathcal{S}^d \cong \mathbb{R}^{nd}$ and $\mathcal{V}_h \cong \mathbb{R}^{nm}$ in our setting. So, discretizing the control by B-splines or NURBS means that any $\Omega = G(\hat{\Omega})$ can be expressed as $\Omega(\mathbf{X})$ where $\mathbf{X} \in \mathbb{R}^{nd}$ a coefficient vector of a B-spline parameterization. In other words, any domain is represented by a vector of control points, corresponding to the classical *design variables*.

The Galerkin discretization yields a discrete state $u_h = \sum \mathbf{u}_i N_i \circ G^{-1}$ where the coefficients $\mathbf{u}_i$ are given by the linear equation $\mathbf{A}\mathbf{u} = \mathbf{F}$ derived from the projected weak form (4.44). In terms of $\mathbf{X}$ and $\mathbf{u}$ and with the discrete cost function $J_h(\mathbf{X}, \mathbf{u}) := J(\hat{u}, \Omega(\mathbf{X}))$, we formulate the discrete isogeometric optimization problem as

$$\min J_h(\mathbf{X}, \mathbf{u}) \quad \text{subject to } \mathbf{A}\mathbf{u} = \mathbf{F}. \tag{4.76}$$

For an optimal pair $(\mathbf{X}, \mathbf{u})$ the necessary first order optimality conditions are

(i) State equation: $\mathbf{A}\mathbf{u} = \mathbf{F}$,

(ii) Stationary point: $\nabla_{\mathbf{X}} J_h(\mathbf{X}, \mathbf{u}) = 0$.

Because $\mathbf{u}$ depends on $\mathbf{X}$ implicitly, we have to take the derivative $\partial_{\mathbf{X}}\mathbf{u}$ for the stationary point condition into account. For this purpose we introduce the index pair $\alpha = (i, k)$ where $i = 1, \ldots, n$ runs over the control points and $k = 1, \ldots, d$ over their components. By $\mathrm{d}_\alpha := \mathrm{d}_{\mathbf{X}_\alpha}$ and $\partial_\alpha := \partial_{\mathbf{X}_\alpha}$ we denote the derivatives with respect to component k of control point X_i. Then by the chain rule

$$\mathrm{d}_\alpha J_h(\mathbf{X}, \mathbf{u}) = \partial_\alpha J_h + \partial_{\mathbf{u}} J_h \cdot \partial_\alpha \mathbf{u} \tag{4.77}$$

where the shape derivative of $\mathbf{u}$ respectively u_h is given by

$$\mathbf{A}\,\partial_\alpha \mathbf{u} = \partial_\alpha \mathbf{F} - \partial_\alpha \mathbf{A}\mathbf{u}. \tag{4.78}$$

We introduce the adjoint $\mathbf{p}$ as the solution of $\mathbf{A}^\mathsf{T}\mathbf{p} = \partial_\mathbf{u} J_h$ and obtain

$$\mathrm{d}_\alpha J_h(\mathbf{X}, \mathbf{u}) = \partial_\alpha J_h + \mathbf{p}^\mathsf{T}(\partial_\alpha \mathbf{F} - \partial_\alpha \mathbf{A}\mathbf{u}). \tag{4.79}$$

Summing up, we have the necessary optimality conditions for an optimal pair $(\mathbf{X}, \mathbf{u})$: For all tuples $\alpha = (i, k)$

$$\mathbf{A}\mathbf{u} = \mathbf{F}, \tag{4.80}$$
$$\mathbf{A}^\mathsf{T}\mathbf{p} = \partial_\mathbf{u} J_h, \tag{4.81}$$
$$\mathrm{d}_\alpha J_h(\mathbf{X}, \mathbf{u}) = \partial_\alpha J_h + \mathbf{p}^\mathsf{T}(\partial_\alpha \mathbf{F} - \partial_\alpha \mathbf{A}\mathbf{u}) = 0. \tag{4.82}$$

4.4.3 Two discretization concepts

Since we claim that the fully discrete systems are equivalent for both the approaches, what makes the difference? First of all, from a theoretical point of view, to show existence and uniqueness of, and convergence to optimal solutions, we need the continuous case, i.e., the *optimize first* part. From a practical angle, some problems require a different ansatz space for the adjoint in (4.63) than $\hat{\mathcal{V}}$, see [Hinze et al., 2009], which also is in favor of *optimize first*. Thus it appears that optimize first is more general. However, for more complicated problems the sensitivities in the Fréchet-sense might not exist and only have numerical interpretations. So both approaches are important, but since we are more interested in the properties of IGA shape optimization, we concentrate here on the optimize first ansatz.

Comparison of the two discretization schemes in IGA

Whichever ansatz one prefers, eventually it yields the same discrete optimality system in IGA for our class of problems. For finite-dimensional Banach spaces the directional derivative of j_i in direction $h_\alpha = N_i \circ G^{-1} e_k$ is just the partial derivative $\partial_\alpha j_i \circ G^{-1}$. That is why for Galerkin methods it does not matter if we use a *discretize first–optimize then* or *optimize first–discretize then* ansatz. We present this result from our publication [Fußeder et al., 2015] in the following theorem.

Theorem 4.34.
Assume that G satisfies (2.31) *with $\Omega = G(\hat{\Omega})$ and that the discretization space for the adjoint in equation* (4.63) *is $\hat{\mathcal{V}}_h$. Then, the partial derivatives w.r.t. shape obtained in the* discretize first–optimize then *and those from the* optimize first–discretize then *ansatz are equal in IGA.*

Proof. Equation (3.55) yields the discrete state equation $\mathbf{A}\mathbf{u} = \mathbf{F}$ of (4.80) and equation (3.56) the discrete adjoint $\mathbf{A}^\mathsf{T}\mathbf{p} = \partial_\mathbf{u} J_h$ in (4.81).

Consider a shape functional without implicit dependency on the domain

$$\begin{aligned} J(\Omega) &= \int_\Omega \phi(x)\,\mathrm{d}x, \qquad \phi\colon \Omega \to \mathbb{R}, \\ &= \int_{\hat{\Omega}} \phi \circ G |\det \mathrm{J}_G|\,\mathrm{d}\hat{x}. \end{aligned} \tag{4.83}$$

In IGA we set $\partial_{X_\alpha} =: \partial_\alpha$ for an arbitrary control point X_α for some $\alpha = (i,k)$, $i = 1,\dots,n$ and $k = 1,\dots,d$. Then,

$$\partial_\alpha J_h(\mathbf{X}) = \int_{\hat\Omega} \Big((\nabla\phi)\circ G \cdot \partial_\alpha G + \phi \circ G \operatorname{tr}(\mathrm{J}_G^{-1}\,\partial_\alpha \mathrm{J}_G) \Big) |\det \mathrm{J}_G|\,\mathrm{d}\hat{x} \tag{4.84}$$

from the chain rule on the transported shape function in (4.83). On the other hand, with $h = N_\alpha \circ G^{-1}$ and Lemma 3.24 we have

$$\mathrm{d}J(G;N_\alpha) = \int_{\hat\Omega} \Big((\nabla\phi)\circ G \cdot N_\alpha + \phi \circ G \operatorname{tr}(\mathrm{J}_G^{-1}\,\mathrm{D}N_\alpha) \Big) |\det \mathrm{J}_G|\,\mathrm{d}\hat{x}. \tag{4.85}$$

From direct calculation it can be seen that $\partial_\alpha G = \partial_\alpha \sum X_{\alpha_i} N_{\alpha_i} = N_\alpha$ and $\partial_\alpha \mathrm{J}_G = \mathrm{D}N_\alpha$, hence $\mathrm{d}J(G;N_\alpha) = \partial_{X_\alpha} J_h(\mathbf{X})$.

Because in $\hat{l}(G)(\hat{v})$ as well as $\hat{a}(G)(\hat{u},\hat{v})$ functions $\hat{u}$, $\hat{v}$ are independent of G, the same arguments as for equations (4.84) and (4.85) apply component-wise:

$$\partial_\alpha \mathbf{F}_{\alpha_i} = \mathrm{d}\hat{l}_G(N_{\alpha_i};N_\alpha)$$

and

$$\partial_\alpha \mathbf{A}_{\alpha_i,\alpha_j} = \mathrm{d}a_G(N_{\alpha_i},N_{\alpha_j};N_\alpha).$$

Therefore,

$$\begin{aligned} \mathrm{d}_\alpha J_h(\mathbf{X}) &= \partial_\alpha J_h + \mathbf{p}^\mathsf{T}(\partial_\alpha \mathbf{F} - \partial_\alpha \mathbf{A}\mathbf{u}), \\ \mathrm{d}J(\Omega;N_\alpha) &= \partial_\alpha J_h + \sum_j \mathbf{p}_j\,\partial_\alpha \mathbf{F}_j - \sum_{i,j} \mathbf{u}_i\,\partial_\alpha \mathbf{A}_{ij}\mathbf{p}_j, \end{aligned}$$

and the discrete systems are equivalent. □

Simple 1D test

In the following shape optimization problem without PDE we use a NURBS space for optimization. The problem is simple enough to differentiate directly w.r.t. the weights and control points. We compare these directly obtained shape gradients to the ones using homogeneous coordinates from above.

Example 4.5 (One-dimensional test case)**.** Let $\phi = \frac{1}{1+x}$ be a function which we would like to track with a NURBS curve on $[0,1]$, i.e. we consider the cost functional

$$J(G) = \|\phi - G\|^2_{L^1(0,1)} = \int_0^1 (\phi - G)^2\,\mathrm{d}x, \tag{4.86}$$

where the geometry is a NURBS curve given by

$$G = \frac{X_0\omega_0 N_0 + X_1\omega_1 N_1}{\omega_0 N_0 + \omega_1 N_1}, \tag{4.87}$$

with positive weights. The latter can be enforced in the optimization routine with inequality constraints $g = \begin{pmatrix} -\omega_0 \\ -\omega_1 \end{pmatrix} < 0$.

The approximation space for G is $\mathcal{N} = \{\mathrm{H}(\tilde{G})\colon \tilde{G} = (G^w, w) \in \mathcal{S}^1 \oplus \mathcal{S}\}$ with a B-spline basis $\mathcal{B}(\Xi, 1) = \{N_0 = 1 - \hat{x}, N_1 = \hat{x}\}$ on the knot vector $\Xi = (0, 0, 1, 1)$ for the spline space $\mathcal{S} = \operatorname{span}\mathcal{B}$. One can check directly that $\omega_0 = 1$, $\omega_1 = 2$, $X_0 = 1$, $X_1 = 0.5$ is the exact solution of Example 4.5, or in other words $\Omega^\star = \phi$. The shape sensitivities are formed by

$$\begin{pmatrix} \nabla_0 J \\ \nabla_1 J \\ \nabla_2 J \\ \nabla_3 J \end{pmatrix} := \begin{pmatrix} \mathrm{d}J(\mathrm{H}(\tilde{G}); N_0 e_1) \\ \mathrm{d}J(\mathrm{H}(\tilde{G}); N_0 e_2) \\ \mathrm{d}J(\mathrm{H}(\tilde{G}); N_1 e_1) \\ \mathrm{d}J(\mathrm{H}(\tilde{G}); N_1 e_2) \end{pmatrix} \stackrel{?}{=} \begin{pmatrix} \partial_{X_0} J \\ \partial_{\omega_0} J \\ \partial_{X_1} J \\ \partial_{\omega_1} J \end{pmatrix}$$

and should correspond to the gradient obtained by directly differentiating J. Exemplary we give the derivative w.r.t. weight ω_0

$$\partial_{\omega_0} J(G) = \int_0^1 -2(\phi - G)\left(\frac{X_0 N_0}{w} - \frac{G}{w} N_0\right) \tag{4.88}$$

$$\mathrm{d}J(G; N_0 e_2) = \int_0^1 -2(\phi - G)\left(-\frac{G}{w} N_0\right). \tag{4.89}$$

The missing term in shape derivative (4.89) is compensated by a different update rule. Namely, compare updating with direct gradient

$$G_{new} = \frac{(X_0 + \partial_{X_0} J)(\omega_0 + \partial_{\omega_0} J) N_0 + (X_1 + \partial_{X_1} J)(\omega_1 + \partial_{\omega_1} J) N_1}{(\omega_0 + \partial_{\omega_0} J) N_0 + (\omega_1 + \partial_{\omega_1} J) N_1} \tag{4.90}$$

and with shape gradient

$$\begin{aligned} \tilde{G}_{new} &= \begin{pmatrix} X_0\omega_0 + \nabla_0 J \\ \omega_0 + \nabla_1 J \end{pmatrix} N_0 + \begin{pmatrix} X_1\omega_1 + \nabla_2 J \\ \omega_1 + \nabla_3 J \end{pmatrix} N_1 \quad \Rightarrow \\ G_{new} &= \frac{(X_0\omega_0 + \nabla_0 J) N_0 + (X_1\omega_1 + \nabla_2 J) N_1}{(\omega_0 + \nabla_1 J) N_0 + (\omega_1 + \nabla_3 J) N_1} \end{aligned} \tag{4.91}$$

As expected, if we approximate ϕ with piecewise linear functions, that is we use no weights in Example 4.5, the error $\epsilon = J(G) - J(\Omega^\star) = J(G)$ is large, $\epsilon \approx 10^{-4}$, compared to a G obtained with simultaneous weight optimization, which is $\epsilon \approx 7 \times 10^{-15}$ in Figure 4.10.

Comparison reviewed from the optimal control point of view

In [Hinze et al., 2009, Ch. 3] the two discretization approaches are compared for usual optimal control problems like (2.44). The result is similar, namely that they lead to the same discrete systems for Galerkin methods if the finite-dimensional function spaces of state and adjoint are the same in *optimize first*.

It seems that the shape optimization community is split into two camps about these approaches. Moreover, the different groups are further divided by computing the shape gradients also by finite differences schemes or automatic differentiation. Of course, each method has its own advantage; comparisons for structural design sensitivities are for

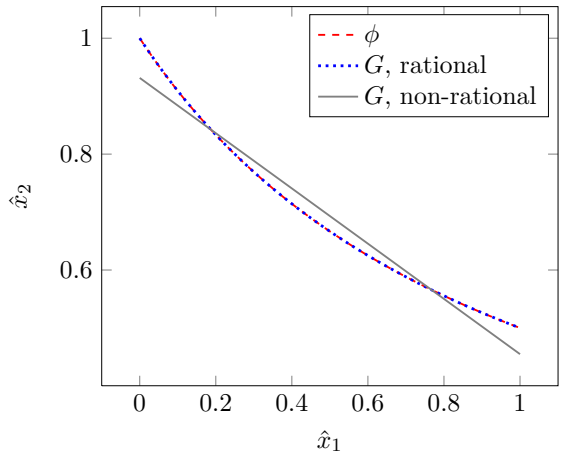


Figure 4.10: Optimization with B-splines and NURBS in Example 4.5: With B-splines the optimal control points are $X_0 = 0.9315$, $X_1 = 0.4548$, whereas for NURBS we get optimal weights and control points up to errors $\epsilon_{\omega_0} = 0$, $\epsilon_{\omega_1} = 1.292 \times 10^{-6}$, $\epsilon_{X_0} = 1.74 \times 10^{-7}$ and $\epsilon_{X_1} = 7.4 \times 10^{-8}$, with starting points $X_0 = 0 = X_1$ and $\omega_0 = 1 = \omega_1$.

instance found in [van Keulen et al., 2005] and [Neittaanmäki and Salmenjoki, 1989]. One main point of argumentation from the *discretize first* side is that due to discretization errors the shape gradient from an *optimize first* view does not fit the numerical optimization problem that is actually solved. In [Berggren, 2010], this discrepancy is addressed with the result that for smooth enough state and adjoint variables the differences can vanish under some assumptions. Though being a common statement in optimal control theory for Galerkin methods, the equivalence is often not even considered in shape optimization. This indifference towards alternative formulations probably roots in the two extremes found in shape optimization: at one end industrial relevant problems where shape optimization theory is not applicable in full rigor, and on the other side mathematical theory of shape calculus, existence, uniqueness and convergence for often only academic problems.

5 Shape Optimization Methods

In Chapter 3 the foundation of shape optimization was laid and in Chapter 4 a discretization of it was discussed. However, for a successful implementation this is not enough yet. Up to now a particular optimization routine is missing that gives us a descent direction based upon information from the (shape) gradient. Since we have constraints on the control in all of our examples we need a solver for constrained nonlinear optimization problems. Due to the constraints, a simple gradient-descent together with a line search for the step size, like the Armijo rule, does not do the job in our cases. Hence, we discuss our choices of optimization solvers in **Section 5.1**. Another point, irrelevant in the theoretical discussion of SOPs, is the construction of new geometries after an optimization step. Such updates of a geometry with a descent direction are something of a bottleneck in shape optimization which deserves a careful investigation. In **Section 5.3** we therefore sketch the problem of infeasible parameterizations and how we try to avoid them in our examples. A global answer to this is out of scope for this thesis; we content ourselves with heuristic approaches. Another important topic in shape optimization is of course the influence of discretization errors. In particular, the well-studied error of PDE approximations enters the SOP solution. Specifically in isogeometric analysis we have the same discretization parameter for optimization and for simulation since control and state space are equal. In order to separate (and thus control) the two sources of discretization errors we loosen the tight relation between geometry, control and state in **Section 5.4**, and show that two different spline spaces for optimization and simulation are supported by our theory.

5.1 Nonlinear Optimization Programs

To solve shape optimization problems like (1.1) after discretization by either *discretize first* or *optimize first* leads to a finite-dimensional problem looking like

$$\min_{x\in\mathbb{R}^N} f(x) \quad \text{subject to} \quad g(x) \leq 0, \quad h(x) = 0 \tag{5.1}$$

with M (nonlinear) inequality constraints $g\colon \mathbb{R}^N \to \mathbb{R}^M$ and P (nonlinear) equality constraints $h\colon \mathbb{R}^N \to \mathbb{R}^P$. In the shape optimization context, the cost function f corresponds to a discrete version of $J(\Omega, u)$, g and h supply constraints on the control Ω, for instance constant volumes. In order to accommodate the restrictions on x in an optimization method the Lagrange function is employed, as in the infinite dimensional problems. Again, optimal $x^\star$ satisfy a Karush-Kuhn-Tucker system of optimality criteria, and optimization methods iteratively optimize the Lagrange function. An overview on numerical minimization methods for shape optimization is given for instance in [Haslinger and Mäkinen, 2003], general optimization algorithms are treated in [Nocedal and Write, 2006] and we follow [Ulbrich and Ulbrich, 2012] in our exposition.

5.1.1 Optimality criteria

Before we state the KKT system for problem (5.1) we introduce some notation. In the following we assume that f, g and h are continuously differentiable and $x \in \mathbb{R}^N$ is feasible if it satisfies the constraints. Furthermore, we denote the *index set of active inequality constraints* by

$$\mathcal{A}(x) := \{i \colon 1 \leq i \leq M,\, g_i(x) = 0\}\,. \tag{5.2}$$

A vector $v \in \mathbb{R}^M$ is reduced to its active entries by the notation $v_{\mathcal{A}(x)} = (v_i)_{i \in \mathcal{A}(x)}$. At an optimal point a so called *constraint qualification* must be satisfied. There are several equivalent conditions to ensure that and here we pick the *linear independence constraint qualification*.

Definition 5.1. A feasible $x \in \mathbb{R}^N$ is called *regular point* if the columns of the matrix $(\nabla g_{\mathcal{A}(x)}, \nabla h)$ are linearly independent. This regularity condition is also called *linear independence constraint qualification* for an optimal point $x^\star$.

For other, especially weaker, constraint qualifications we refer to literature. To formulate the KKT optimality system for the finite-dimensional problem (5.1) we need (as in the infinite case) couple the objective and constraints in the *Lagrange function*.

Definition 5.2. For problem (5.1) the Lagrange function is defined as

$$\mathcal{L}(x, \lambda, \mu) := f(x) + \lambda^\mathsf{T} g(x) + \mu^\mathsf{T} h(x) \tag{5.3}$$

where $\lambda \in \mathbb{R}^M$ and $\mu \in \mathbb{R}^P$.

We now state from [Ulbrich and Ulbrich, 2012, p. 94] a necessary first order optimality condition for constrained nonlinear optimization problems.

Theorem 5.3 (Necessary optimality criteria of first order).
Let $x^\star \in \mathbb{R}^n$ be a local solution to the finite-dimensional optimization problem (5.1) *where a constraint qualification is fulfilled. Then the* Karush-Kuhn-Tucker *conditions are true: There exist* Lagrange multipliers *$\lambda^\star$ and $\mu^\star$ such that*

(i) $\nabla_x \mathcal{L}(x^\star, \lambda^\star, \mu^\star) = 0$,
(ii) $\nabla_\mu \mathcal{L}(x^\star, \lambda^\star, \mu^\star) = h(x^\star) = 0$,
(iii) $\nabla_\lambda \mathcal{L}(x^\star, \lambda^\star, \mu^\star) = g(x^\star) \leq 0,\ \lambda^\star \geq 0,\ \lambda^{\star\mathsf{T}} g(x^\star) = 0$.

5.1.2 Nonlinear Programming Algorithms

For the finite-dimensional nonlinear optimization problem (5.1) we use gradient-based methods. In particular, we use sequential quadratic programming, method of moving asymptotes, and an interior point method to solve the applications in Chapter 6. Although all of them use the gradient for updating and checking optimality criteria, they differ in their strategies. Among many others, [Nocedal and Write, 2006] offers an extensive treatment of numerical optimization methods.

Iterative optimization methods

Iterative optimization methods for (5.1) produce a sequence of updates

$$x_{k+1} = x_k + t_k s_k \tag{5.4}$$

where in the kth step s_k is a descent direction and t_k a step size. To begin the iteration one has to chose a starting point x_0 as initialization.

Unconstrained optimization

We look at unconstrained minimization problems to motivate the different descent methods.

Steepest descent A standard *steepest descent method* sets the descent direction to $s_k = -\nabla f(x_k)$ and the step size t_k is determined for instance by an *Armijo rule:* For a $\gamma \in (0,1)$

$$\text{test for } t = 2^{-\ell},\ \ell = 0,\ 1,\ 2,\ \ldots \text{ whether } f(x_k + t s_k) - f(x_k) \le t\gamma\, \nabla f(x_k)^\mathsf{T} s_k\,. \tag{5.5}$$

Set t_k to the largest such t. This can be rather slow with only *linear convergence* to an $x^\star$, i.e. there is a constant $0 < C < 1$ and $N \ge 0$ with

$$\frac{\|x_{k+1} - x^\star\|}{\|x_k - x^\star\|} \le C \quad \forall k \ge N\,. \tag{5.6}$$

Newton method In contrast, Newton methods converge faster by using information of second derivatives. If f is twice continuously differentiable then a Taylor expansion yields $f(x_k + s) = f(x_k) + \nabla f(x_k)^\mathsf{T} s + \frac{1}{2} s^\mathsf{T} \nabla^2 f(x_k) s + o(\|s\|^2)$. Therefore, to minimize f we minimize $f(x_k + s)$ w.r.t. s which is the same as finding the minimum s_k to the quadratic term $F(s) := \nabla f(x_k)^\mathsf{T} s + \frac{1}{2} s^\mathsf{T} \nabla^2 f(x_k) s$. The first order necessary optimality condition for F, i.e. $\nabla F(s_k) = 0$, leads to the Newton equation for $\nabla F = \nabla^2 f$ and thus to a linear system for the descent direction

$$\nabla^2 f(x_k) s_k = -\nabla f(x_k) \tag{5.7}$$

and $t_k = 1$. Under appropriate conditions the Newton method converges

$$\textit{superlinearly:}\ \frac{\|x_{k+1} - x^\star\|}{\|x_k - x^\star\|} \to 0 \text{ for } k \to \infty\,, \text{ or even} \tag{5.8}$$

$$\textit{quadratically:}\ \frac{\|x_{k+1} - x^\star\|}{\|x_k - x^\star\|^2} \le C \quad \forall k \ge 0\,. \tag{5.9}$$

Constrained optimization

We infer the motivation from the unconstrained case above also to the constrained optimization.

Sequential Quadratic Programming (SQP) methods are one of the most efficient optimization methods [Ulbrich and Ulbrich, 2012, p. 120]. They compute the descent direction for the update formula (5.4) by solving the following quadratic sub problems where $H_k \approx \nabla^2_{xx} \mathcal{L}(x_k, \lambda_k, \mu_k)$

$$\begin{aligned} \min_{s \in \mathbb{R}^n} \nabla f(x^k)^\mathsf{T} s + \frac{1}{2} s^\mathsf{T} H_k s \quad \text{s.t.} \\ g(x^k) + \nabla g(x^k)^\mathsf{T} s \leq 0\,, \\ h(x^k) + \nabla h(x^k)^\mathsf{T} s = 0\,. \end{aligned} \tag{5.10}$$

Under appropriate conditions the SQP algorithm converges q-superlinearly or even q-quadratic if $H_k = \nabla^2_{xx} \mathcal{L}(x_k, \lambda_k, \mu_k)$. Without second derivative information one uses so called quasi-Newton–updates to approximate H_k. For such updates we refer to [Ulbrich and Ulbrich, 2012] and others, but remark that even with these approximations we can achieve better convergence rates than linear ones.

Method of Moving Asymptotes (MMA) is an optimization method which was developed for shape optimization, particularly in structural mechanics and is attractive for problems with a large number of design variables. The rough idea is to replace f, g and h in each iteration by a linearization through a Taylor expansion around a cleverly chosen point. This expansion point is adjusted by parameters, the moving asymptotes, which are the new optimization variables. Thus one ends up with dual sub problems which are much cheaper to solve than the primal problem. For the formulation of these sub problems and a proof of convergence we refer to the inventor [Svanberg, 2002]. However, let us remark that MMA does not support equality constraints and that SQP typically has a faster convergence rate.

Interior Point Method These methods handle large-scale nonlinear problems with inequality constraints. They reformulate the problem by adding a logarithmic barrier to $f(x)$ that penalizes movements towards the boundary of the feasible set

$$\min_{x \in \mathbb{R}^N} f_\mu := f(x) - \mu \sum_{i=1}^{N} \ln(x^i)\,. \tag{5.11}$$

We substitute f_μ for f in (5.1) and the method in [Wächter and Biegler, 2006] solves this barrier problem for a fixed μ and then decreases the barrier parameter successively. We refer to [Nocedal and Write, 2006] for an introduction or, for details to the `IPOPT` implementation, to [Wächter and Biegler, 2006].

5.2 Sobolev Smoothing

Though both interpretations of directional derivatives, *optimize first* and *discretize first*, lead to the same system for the directional derivative, in the functional space setting the gradient is a member of a Hilbert space and thus depending on the underlying scalar product $(\cdot, \cdot)$. For a Fréchet-differentiable real-valued function J, the gradient denoted by $\nabla J \in H$ is according to (3.24) the solution to

$$(\nabla J, \theta) = \mathrm{d}J(G; \theta) \quad \forall \theta \in H\,. \tag{5.12}$$

This leaves some freedom in choosing the inner product $(\cdot, \cdot)$ and hence the regularity of ∇J. According to [Protas et al., 2004] and [Schmidt et al., 2011] such a regularization has the effect of preconditioning the optimization process. As yet we have not used this *Sobolev-smoothing* of the gradient in any of the model problems. However, in [Blanchard et al., 2013] an IGA example can be found that combines regularization with mesh movement from Section 5.3.

5.3 Mesh Update Strategies

Large deformation at the boundary of a geometry due to updates by shape gradients frequently leads to infeasible parameterizations, for instance, when the control polygon overlaps with inner control lines, as in Figure 5.1b. Such tangled meshes also are a curse of time evolving domains as in flow problems with moving boundaries, and likewise in shape optimization. The measures to avoid them depend on the "nature and magnitude" of the desired changes, [Johnson and Tezduyar, 1994], a last resort is completely remeshing the new domain. The latter is no option for isogeometric shape optimization because there is no simple isogeometric mesh generator yet. In fact, obtaining analysis suitable volume models from surface parameterizations is an ongoin research process, [Jüttler et al., 2014]. However, not all deformations have this fatal flaw: small deformations or those in direction of the mesh structure are more benign. So, we adapt our measures as situation demands. Since it is a purely discrete problem we first formulate it as such in a general way, to be filled later with meaning by identified update strategies. After commenting on methods already employed in isogeometric shape optimization, we describe the ones we pick for this work.

(a) Initial parameterization with moving boundary $\hat{\Gamma}$

(b) Overlapping parameterization after moving only the boundary

Figure 5.1: Moving only the boundary can cause irregular parameterizations.

5.3.1 General mesh movement

We write here very generally that a mesh movement function $\Phi\colon \mathbb{R}^{n_{des}} \to \mathbb{R}^{nd}$ propagates a descent direction s to all control points, where s results from the sensitivity analysis and has the size n_{des} corresponding to the number of design variables. The focus is here on an algorithmic framework for shape optimization, therefore we do not specify Φ now or for our shape optimization algorithm. However, we discuss subsequently some option how to fill Φ with meaning. This general approach is justified because in the end an update in IGA is carried out by changing the control points.

We say that a function $\Phi(s) = y \in \mathbb{R}^{nd}$ propagates the descent direction to all control points, respecting fixed boundaries and the movement on the optimization boundary, i.e.

$$\begin{cases} y_\ell = s_\ell & \text{if } N_{i_\ell} e_{k_\ell} \text{ is design variable} \\ y_\ell = 0 & \text{if } N_{i_\ell} e_{k_\ell} \text{ is on fixed boundary}\,. \end{cases} \tag{5.13}$$

Existing mesh movement methods comprise techniques from different fields like flow problems [Johnson and Tezduyar, 1994] and r-adaptivity [Budd et al., 2009], but are also taylored towards shape optimization [Hicken and Zingg, 2008]. However, they seem to be problem specific in any case.

5.3.2 Mesh update in isogeometric shape optimization

In IGA shape optimization mainly two methods have been used so far to move inner control points, solving another optimization problem, or using linear elasticity. We comment our options.

Mesh movement by solving another optimization problem

For an electromagnetic shape optimization in [Nguyen et al., 2012], the movement of inner control points is realized by minimizing the *Winslow functional* which aims for a positive determinant of the Jacobian of G. However, solving an additional optimization problem can become quite expensive, therefore the authors suggest a linearized version. Closely related to this method are the analysis-aware IGA meshes in [Xu et al., 2013]. There, the authors improve the position of inner control points such that the error in PDE analysis decreases. This is done by formulating the problem as an SOP and as such also optimizes the spread of the magnitude of the determinant of the Jacobian of G. However, for this technique a good error estimator is needed.

Mesh movement with PDEs

In general, finding a mesh movement vector field Φ can be formulated as another PDE as shows [Blanchard et al., 2013], where a mesh deformation is realized by solving a linear elasticity problem with the displacement of control points on the moving boundary as Neumann boundary condition. This is especially attractive in linear elasticity state equations since then the system matrix of the state can be used in the mesh deformation problem as well. Note however, that the resulting mesh movement depends then on the parameters of the elasticity equation. Then, a "small" shape gradient might effect little or no mesh movement for rigid materials.

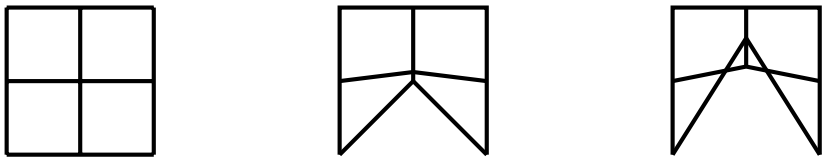
For a Poisson state equation we show with a simple example that a mesh update by the Poisson stiffness matrix can fail if the step size in an optimization loop is too large and leads to non-convex domains. We construct a toy example: We use the Poisson equation with inhomogeneous Dirichlet boundary conditions to move the mesh

$$-\Delta\,\Phi = 0 \text{ in } (0,1)^2, \quad \Phi = f_t(x) \text{ on } [0,1]\times\{0\}, \text{ and } \Phi = 0 \text{ else on } \partial(0,1)^2 \tag{5.14}$$

where for $t \in [0, 1)$ the deformed boundary is given as

$$f_t(x) = \begin{cases} tx & x \leq 0.5\,, \\ 1 - tx & x > 0.5\,. \end{cases}$$

This yields irregular parameterizations for larger step sizes as shown in Figure 5.2.



(a) $t = 0$ **(b)** $t = 0.5$ **(c)** $t = 0.8$

Figure 5.2: We show with a minimal example that moving the inner control points by solving a Poisson equation may yield irregular parameterizations if changes to the domain are too large and non-convex. We use the knot vector $\Xi = (0, 0, 0.5, 1, 1)$ in both directions for B-splines of degree $p = 1$ and variable second control point $(0.5, t)^\intercal$ with $0 \leq t < 1$. The initial configuration is given at $t = 0$. For $t = 0.5$ we have an admissible step size, $t = 0.8$ is already too large.

Mesh movement based on the distance to the boundary

Another alternative is to use a relative positioning of inner control points which is quite easy to implement in IGA due to the tensor product space structure: In $d = 2$ dimensions, let the boundaries of Ω be given by functions γ_N, γ_S, γ_W and γ_E which are B-spline or NURBS parameterizations. For instance $\gamma_S = G(\hat{x}_1, 0)$, $\gamma_N = G(\hat{x}_1, 1)$, $\hat{x}_1 \in (0, 1)$, and so forth. Then

$$T(\hat{x}_1, \hat{x}_2) = id + \begin{pmatrix} \gamma_W(\hat{x}_2)(1 - \hat{x}_1) - \hat{x}_1(1 - \gamma_E(\hat{x}_2)) \\ \gamma_S(\hat{x}_1)(1 - \hat{x}_2) - \hat{x}_2(1 - \gamma_N(\hat{x}_1)) \end{pmatrix} \tag{5.15}$$

transforms $\hat{\Omega}$ to Ω. In particular, we get equidistantly spaced control points by mapping the intersecting grid points of $\hat{\Omega}$ by means of T to Ω. The method works if control points only move up and down, examplary the movement from Figure 5.1 results with this method in Figure 5.3. However, it works only to some extent if control points move diagonally as in Figure 5.4.

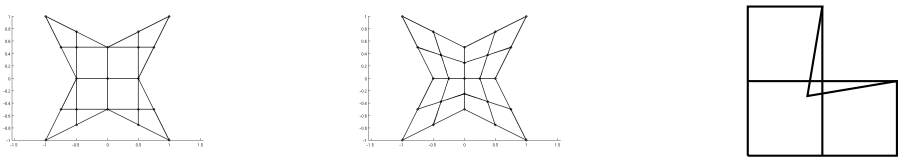
5.4 Decoupling of State and Control Discretization

Consider Remark 4.27 where we assumed two spline spaces $\mathcal{S}_1$, $\mathcal{S}_2$: one for the geometry representation and one for the projection space. Since optimization is very costly for each design variable we want to keep the number of design variables as low as design considerations allow. However, the interpolation error of the state equation directly plays a role in evaluating the objective function J and also in the accuracy of its gradient, which in turn influences the optimization. As an intuitive example, suppose we solve the Poisson equation on the unit square with homogeneous Dirichlet boundary conditions with only two linear basis functions per direction, then $u_h = 0$,

(a) Overlapping parameterization after moving only the boundary

(b) Parameterization by relative positioning of inner control points

Figure 5.3: Adjusting inner control points by relative positioning to avoid mesh tangling: Instead of moving only the boundary, we use transformation T of (5.15) to obtain an equidistant parameterization



(a) Deformed boundaries

(b) Relative positioning works here for moving inner control points

(c) Relative positioning fails

Figure 5.4: An example where relative positioning of inner control points works in non-convex domains and one where it does not

which leaves not much scope for the optimizer. Hence, a good approximation of the state is vital. For detailed studies on the influence of the error $u - u_h$, see for instance [Kiniger and Vexler, 2013, Eppler et al., 2007].

Of course, the interpolation error of design also influences the accuracy of the optimization outcome. Hence using two separate meshes makes it possible to control the sources of discretization errors in design and analysis separately as well.

Both schemes, *optimize first–discretize then* and *discretize first–optimize then* allow for this scenario of two spline spaces. In the Karush-Kuhn-Tucker system (3.55)–(3.57) the variations $\delta\hat{v}$ and $\delta\hat{u}$ are taken from $\hat{\mathcal{V}} = \mathcal{S}_2^m \cap W_0^1(\hat{\Omega})^m$, whereas for domain perturbations one selects $\delta\hat{q} \in \mathcal{S}_1^d$. In the discretize first system (4.80)–(4.82) the shape derivatives ∂_{X_α} refer to control points of $G \in \mathcal{S}_1^d$.

5.5 Local Refinement

With the evolution of IGA to an established PDE solver the ability for adaptive refinement is a crucial step to play in the liga of classic FEM solvers. However, due to the tensor product structure of the B-spline space a local refinement in one dimension propagates the refinement in the other components, see Figure 5.5. There are several variants of B-splines capable of local refinement. In this work we use *hierarchical B-splines* which were developed in [Vuong et al., 2011] and [Vuong, 2012]. The properties and implementation aspects of hierarchical B-splines can be found in the references above. We summarize from there, how to construct a hierarchy of B-splines and how to address

the structure of the corresponding mesh. Finally, we give an example of a locally refined simulation.

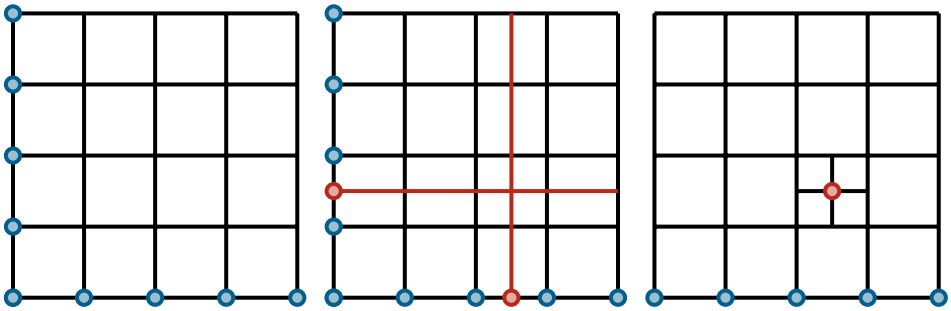


Figure 5.5: Propagation of local refinement: Consider a bivariate B-spline space resulting from knot vector $\Xi = (0, 0, 0.25, 0.5, 0.75, 1, 1)$ for both directions, resulting in the mesh on the left. Inserting a new not at 0.625 in the first direction and at 0.375 in the second propagates the refinement throughout the tensor structure, visible in the mesh in the middle. Opposed to this is a refinement of only one element given by the right mesh, breaking up the tensor product structure.

5.5.1 Construction of hierarchical basis

The aim of this section is to construct a set of linearly independent B-spline basis functions which preserve the tensor product structure and can represent local details of geometries or of PDE solutions. The latter can be achieved by classical hierarchical B-splines from [Forsey and Bartels, 1988]. However, the basis functions may not be linearly independent in this approach, so we need the method from [Vuong et al., 2011] to overcome this. We first define a hierarchy of B-spline spaces and then give a recursion to collect a linearly independent basis from them.

Let $L > 0$ be a finite number of levels and for all $\ell = 1, \dots, L$ define a bivariate B-spline basis $\mathcal{B}_\ell = \mathcal{B}(\Xi_1^\ell, p_1; \Xi_2^\ell, p_2)$ as in Definition 4.4. We assume that the degree stays the same for all levels and that the knot vectors are nested: For all $1 \leq \ell < L$ it holds

$$\begin{cases} \Xi_1^\ell \subset \Xi_1^{\ell+1} \text{ and } \Xi_2^\ell \subset \Xi_2^{\ell+1}, \\ \mu^\ell \leq \mu^{\ell+1} \text{ for the multiplicity } \mu \text{ of any knot in } \Xi^\ell. \end{cases} \tag{5.16}$$

Furthermore, assume there is a sequence of bounded closed sets $(\hat{\Omega}_\ell)_{\ell=1,\dots,L}$ with

$$\hat{\Omega}_L \subset \hat{\Omega}_{L-1} \subset \ldots \subset \hat{\Omega}_1 = \bar{\hat{\Omega}} \tag{5.17}$$

which define the refinement regions. Then we can give a definition of classical, hierarchical B-splines.

Definition 5.4. For $L > 0$, the sequence of nested B-spline spaces $(\mathcal{S}_\ell)_{\ell=1,\dots,L}$ defined over the parameter domain $\hat{\Omega}$ and spanned by the basis $\mathcal{B}_\ell$

$$\mathcal{S}_1 \subset \mathcal{S}_2 \subset \ldots \subset \mathcal{S}_L \tag{5.18}$$

together with the sequence of refinement regions $(\hat{\Omega}_\ell)_{\ell=1,\dots,L}$ define *hierarchical* B*-splines* $\mathcal{B}$: From each level ℓ the basis functions from $\mathcal{B}_\ell$ with support in $\hat{\Omega}_\ell$ are selected for the classical hierarchical basis:

$$\mathcal{B} := \bigcup_{\ell=1}^{L} \{N \in \mathcal{B}_\ell \colon \operatorname{supp} N \subset \hat{\Omega}_\ell\} \,. \tag{5.19}$$

Note that we are guilty of an abuse of notation since $\mathcal{B}$ from Definition 5.4 is not a basis; the B-splines in $\mathcal{B}$ are not linearly independent. In the next step we fix this according to [Vuong, 2012, p. 85 ff.] to define hierarchical B-splines for isogeometric analysis. This is done by removing basis functions from $\mathcal{B}$ which can be expressed by those from the next levels.

Definition 5.5. From the sequences $(\mathcal{S}_\ell)_{\ell=1,\dots,L}$ and $(\hat{\Omega}_\ell)_{\ell=1,\dots,L}$ from Definition 5.4 initialize $B_1 := \mathcal{B}_1$. Then construct the next level from the current one by the recursion for $\ell = 1, \dots, L-1$

$$B_{\ell+1} = (B_\ell \setminus B_\ell^{\ominus}) \cup B_{\ell+1}^{\oplus} \,, \tag{5.20}$$

where

$$B_\ell^{\ominus} = \{N \in B_\ell \colon \operatorname{supp} N \subset \hat{\Omega}_{\ell+1}\} \,, \tag{5.21}$$

$$B_{\ell+1}^{\oplus} = \{N \in \mathcal{B}_{\ell+1} \colon \operatorname{supp} N \subset \hat{\Omega}_{\ell+1}\} \,. \tag{5.22}$$

In the end, set $\mathcal{B} := B_L$.

In Chapter 5 of [Vuong, 2012] the linear independence of hierarchical basis from Definition 5.5 is proven, therefore we can define:

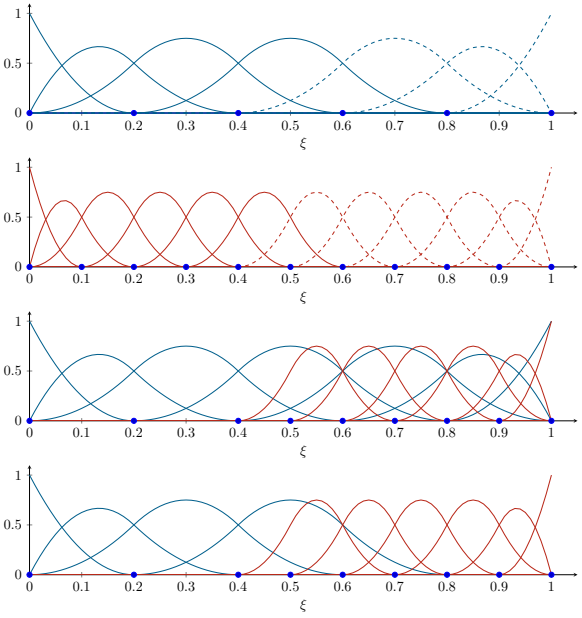
Definition 5.6. We define the *hierarchical B-spline space* as $\mathcal{S}_{hb} := \operatorname{span} \mathcal{B}$.

With an example also from above reference, we illustrate the difference between classical, hierarchical B-splines and hierarchical B-splines for isogeometric analysis in Figure 5.6. Furthermore from this figure, it is visible that not only the basis is locally refined but also the elements or knot spans. In Figure 5.6d are 2 coarse elements on level 1 and 6 refined elements on level 2. A general structure of hierarchical elements is pursued in the next section.

5.5.2 Structure of hierarchical mesh

In the section about isogeometric meshes on page 70 we related B-spline basis functions to elements in the parameter domain by the nonempty knot spans of their corresponding knot vectors. Analogously to adaptive refinement in finite element methods we want to single out elements for local refinement.

We assume that we have $L > 0$ nested spline spaces $\mathcal{S}_\ell$ as in Definition 5.4. Then there is for all spline spaces $\mathcal{S}_\ell$, $\ell = 1, \dots, L$, an isogeometric mesh $\hat{\mathcal{K}}_{h_\ell}$. From these L meshes we select elements $Q \in \hat{\mathcal{K}}_{h_\ell}$ and form a new mesh $\hat{\mathcal{K}}_{hb}$.



(a) Level 1 univariate B-spline basis $\mathcal{B}_1$ of degree $p = 2$ from knot vector $\Xi = (0, 0, 0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1, 1, 1)$

(b) Level 2 univariate B-spline basis $\mathcal{B}_2$ of degree $p = 2$ where each knot span has been subdivided by knot insertion

(c) Classical, hierarchical B-splines with linearly depended spline functions in $\mathcal{B}$

(d) Hierarchical B-splines for isogeometric analysis with linearly independent basis $\mathcal{B}$

Figure 5.6: Classical and isogeometric hierarchical B-splines from Definition 5.4 and Definition 5.5

Definition 5.7. We call $\hat{\mathcal{K}}_{hb} \subset \bigcup_{\ell=1}^{L} \hat{\mathcal{K}}_{h_\ell}$ a *hierarchical isogeometric mesh* if

$$\begin{cases} Q \cap Q' = \emptyset \text{ for all pairwise different } Q,\, Q' \in \hat{\mathcal{K}}_{hb}\,, \\ \bigcup\{\bar{Q} \in \hat{\mathcal{K}}_{hb}\} = \bar{\hat{\Omega}}\,. \end{cases} \tag{5.23}$$

An element $Q \in \hat{\mathcal{K}}_{hb}$ is called *active element* and $\hat{\mathcal{K}}_{hb,\ell} = \hat{\mathcal{K}}_{hb} \cap \hat{\mathcal{K}}_\ell$ is the set of all active elements on level ℓ. The active elements in $\hat{\mathcal{K}}_{hb,\ell}$ cover the region $U_\ell := \bigcup\{\bar{Q} \in \hat{\mathcal{K}}_{hb,\ell}\}$ in the closure of the parameter domain $\bar{\hat{\Omega}}$.

From active elements we obtain active basis functions:

Definition 5.8. Let $N \in \mathcal{B}_\ell$ be a basis function from level ℓ. Then it is *active* if

$$\operatorname{supp} N \subset \bigcup_{k=\ell}^{L} U_k \text{ and } \operatorname{supp} N \nsubseteq \bigcup_{k=\ell+1}^{L} U_k\,. \tag{5.24}$$

The authors in [Vuong, 2012] show that the set of all active basis functions from Definition 5.8 give a hierarchical basis $\mathcal{B}$ from Definition 5.5 with refined regions $\hat{\Omega}_\ell = \bigcup_{k=\ell}^{L} U_k$.

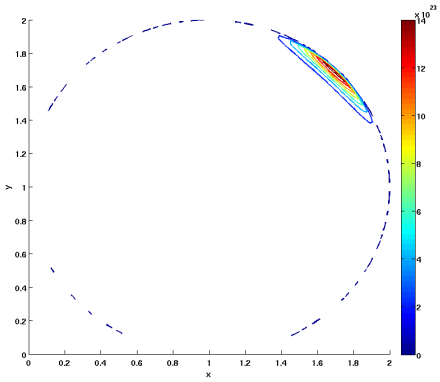
Next, we give an example of simulation with hierarchical B-spline that resolves the local region of interest with the help of an error indicator.

5.5.3 Example of local refinement

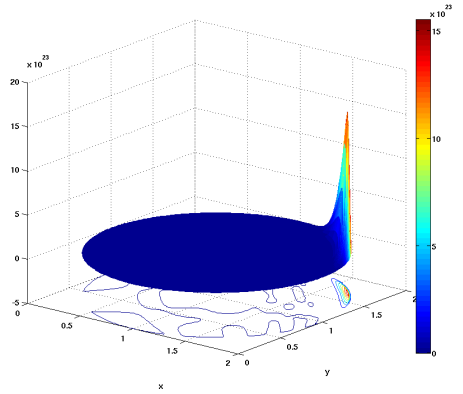
The next example problem modifies the one in our paper [Vuong and Fußeder, 2013] originally from [Prudhomme and Oden, 1999] to apply local adaptive refinement to a PDE defined over a disk.

Example 5.1. We solve the PDE Problem 1, the Poisson equation $-\Delta u = f$ in Ω with homogeneous Dirichlet boundary conditions $u = 0$ on $\partial\Omega$. The right hand side f is chosen such that the equation is fulfilled for $u = -5((x-1)^2 + (y-1)^2 - 1)(\mathrm{e}^{10x^2} - 1)(\exp^{10y^2} - 1)$. The domain $\Omega = \{(x, y) \in \mathbb{R}^2 \colon (x-1)^2 + (y-1)^2 \leq 1\}$ is a disk of radius 1 around the center $(1, 1)$, for which we have a NURBS representation with the data C.3 in the Appendix. Since the data there is for a disk around the origin, we use a translation by adding $(1, 1)$ to all control points.

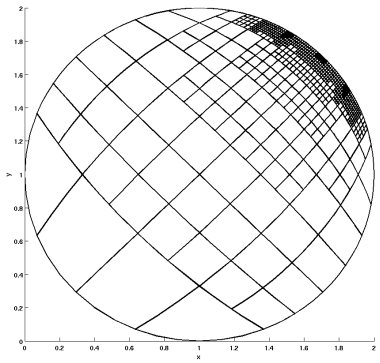
We see from the contour plot of the exact solution in Figure 5.7a that the problem has highly local features which can be reproduced by isogeometric analysis using adaptive refinement in Figure 5.7b. The adaptive refinement with a multilevel error indicator leads to the refined mesh shown in Figures 5.7c and 5.7d with 7 different levels of refinement. Besides local refinement, Example 5.1 exhibits a geometry map with 4 singularities discussed already in Section 4.2.2, Figure 4.7. The sensible results indicate that the regularity conditions on a geometry map may be relaxed as foretold.



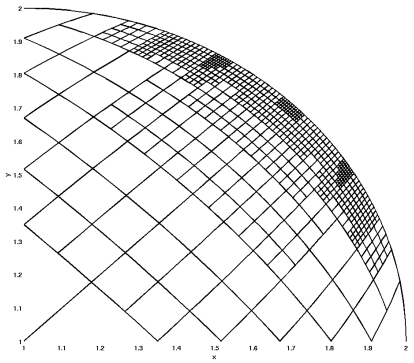
(a) Contour plot of exact solution



(b) Numerical solution and its contours by local refinement



(c) Mesh of physical domain showing 7 levels of refinement



(d) Zooming in of refined mesh

Figure 5.7: Simulation for Example 5.1 with local refinement by hierarchical B-splines

6 Computational Framework and Applications

Finally, in this chapter we tie together the theory and experience from all previous chapters and put it into practice. A gradient-based optimization method is employed to solve the discretized version (5.1) of the minimization problem (1.1) with a black box optimization program, described in Section 5.1.

Using such a black box gradient-based optimization routine we illustrate the steps towards an optimal shape in Algorithm 1.

Algorithm 1 Basic Shape Optimization Algorithm

Require: Initial geometry Ω
Require: PDE(Ω) ▷ PDE solver on Ω which yields solution u_h
Require: OBJECTIVE(Ω, u_h) ▷ evaluate objective function $J(\Omega, u)$ for domains Ω
Require: CONSTRAINTS(Ω)▷ evaluate constraint functions $g(\Omega)$, $h(\Omega)$ for domains Ω
Require: SHAPE GRAD(Ω, u_h) ▷ compute shape gradient ∇J, ∇g, ∇h
Require: UPDATE(Ω, s)▷ update geometry "$\Omega_{new} = \Omega + s$" with descent direction s

Black Box Optimization

repeat
  $u_h \leftarrow$PDE(Ω)
  $J \leftarrow$ OBJECTIVE(Ω, u_h)
  $(g, h) \leftarrow$CONSTRAINTS(Ω)
  $(\nabla J, \nabla g, \nabla h) \leftarrow$SHAPE GRAD($\Omega, u_h$)
  if Ω is not optimal, i.e. does not satisfy a stopping criteria **then**
    compute a descent direction s involving ∇J, ∇g, ∇h
    $\Omega \leftarrow$ UPDATE(Ω, s)
  end if
until Ω is optimal

This algorithmic aspect of isogeometric shape optimization is in the focus of our publication [Fußeder and Simeon, 2015]. Algorithm 1 provides the structure for **Section 6.1**: We apply the theory from the previous sections to obtain the required procedures in Algorithm 1 for the abstract shape optimization problem, and state them explicitly for some model examples. The implementation heavily relies on powerful third party libraries, thankfully acknowledged in this section. We finalize this part with the numerical solution of model problems in **Section 6.2**.

6.1 Isogeometric Shape Optimization Algorithms

On the basis of Algorithm 1 we discuss algorithmic aspects of shape optimization with IGA. In particular, we connect the procedures PDE and SHAPE GRAD to the

shape optimization framework in Chapter 3 and to its discretization with isogeometric analysis from Chapter 4. We briefly describe the third party software we use, especially the optimization solvers, and give credit to the isogeometric PDE solver from my predecessor. With the step-by-step process before us, we try to assess the merit of IGA in shape optimization newly.

6.1.1 PDE solver

Going back to the basic shape optimization Algorithm 1, the procedure PDE(Ω) in line 2 solves the variational form of the state equation of the shape optimization problem, the PDE $e(\Omega, u) = 0$, numerically.

That is done by the means of isogeometric analysis from Section 4.3 which corresponds to assembling and solving the transformed state equation (4.44) where the domain Ω is given by a B-spline or NURBS representation G.

This procedure already acknowledges two discretizations, $\mathcal{S}^d_{geo}$ for the geometry and $\mathcal{S}^m_{sim}$ for the PDE as developed in (4.45) and (4.46). It finally leads to a linear system

$$\mathbf{A}\mathbf{u} = \mathbf{F}, \quad \mathbf{A}_{il,jk} := \hat{a}(G)(M_i e_l, M_j e_k), \quad \mathbf{F}_{il} := \hat{l}(G)(M_i e_l) \tag{6.1}$$

for the model problems with stiffness matrix $\mathbf{A}$, right hand side $\mathbf{F}$ and coefficients of the solution $\hat{u}_h = \sum_{i=1}^{n}\sum_{l=1}^{m} \mathbf{u}_{il} M_i e_l$. We assume that the coefficients $\mathbf{u}_{il} \in \mathbb{R}$ can be ordered lexicographically in a solution vector $\mathbf{u}$, for instance like

$$\mathbf{u} = \begin{pmatrix} \mathbf{u}_{11} & \mathbf{u}_{12} & \dots & \mathbf{u}_{1m} & \dots & \mathbf{u}_{n_{sim}m} \end{pmatrix}^{\mathsf{T}}$$

and in the same breath we write all coefficients belonging to the same spline basis as components of a control point of the solution

$$\mathbf{u}_i = \begin{pmatrix} \mathbf{u}_{i1} & \dots & \mathbf{u}_{im} \end{pmatrix}^{T}.$$

Hence, for any admissible domain Ω given by a B-spline/NURBS parameterization $G = \sum_{i=1}^{n_{geo}} X_i N_i \in \mathcal{S}^d_{geo}$ and a test function space of B-spline/NURBS $\mathcal{S}^m_{sim}$ the procedure PDE(Ω) is given by Algorithm 2. Note, that the assembly can be done element-wise due to the local support of the basis functions, properties (ii) and (iii) in Lemma 4.10 and that the solution u is approximated by $u_h = \hat{u}_h \circ G^{-1}$.

Algorithm 2 PDE(Ω)

1: **for all** basis functions $M_i e_l \in \mathcal{S}^m_{sim}$ **do**
2: **for all** basis functions $M_j e_k \in \mathcal{S}^m_{sim}$ **do**
3: $\mathbf{A}_{il,jk} \leftarrow \hat{a}(G)(M_i e_l, M_j e_k)$
4: **end for**
5: $\mathbf{F}_{il} \leftarrow \hat{l}(G)(M_i e_l)$
6: **end for**
7: Apply boundary conditions of state to $\mathbf{A}$ and $\mathbf{F}$
8: Solve $\mathbf{A}\mathbf{u} = \mathbf{F}$
9: $\hat{u}_h \leftarrow \sum_{i=1}^{n_{sim}} \mathbf{u}_i M_i$.

6.1.2 Shape gradient computation

In this section we bring the shape gradient calculation from Section 4.4 in an algorithmic form. Moreover, we summarize the practical considerations from Chapter 5, Sobolev smoothing, mesh update and decoupled meshes, in one procedure for Algorithm 1.

Here, we make use of compatible meshes from Definition 4.15 in Section 4.1.5: When using two meshes – simulation mesh and optimization mesh – shape sensitivities like $\mathrm{d}_G\hat{a}(G)(M_ie_l, M_je_k; N_re_s)$ occur which involve integrals over basis functions M_ie_l and M_je_k from the simulation space and N_re_s from the geometry space, to vary the domain. Therefore, the domain of integration is the mutual support $\operatorname{supp}(N_r) \cap \operatorname{supp}(M_i) \cap \operatorname{supp}(M_j)$. For an easy implementation, it is favorable when the meshes match, i.e. if they are compatible in the sense of our Definition 4.15. We then can assemble over the elements from the finer mesh and the mutual support is found straight forwardly as shown in Lemma 4.17. Algorithm 3 exploits this procedure to calculate the shape gradient.

Algorithm 3 SHAPE GRAD(Ω, u)

Shape gradient of objective function ▷ **compute terms in** (3.57) **for objective function**

for all basis functions M_ie_l **do** ▷ assemble right hand side of adjoint equation
  $B_{i,l} \leftarrow \mathrm{d}_u\hat{J}(G, M_ie_l)$
end for
Solve $\mathbf{A}^\mathsf{T}\mathbf{p} = B$: adjoint state $\hat{z}_h = \sum_{i=1}^{d} \mathbf{p}_i M_i$ ▷ solve adjoint
for all design variables N_re_s **do** ▷ directional derivatives of objective function
  $\nabla J_{r,s} \leftarrow \mathrm{d}_{\hat{q}}\hat{J}(G, \hat{u}_h; N_re_s) + \mathrm{d}_{\hat{q}}\hat{l}(G)(\hat{z}_h; N_re_s) - \mathrm{d}_{\hat{q}}\hat{a}(G)(\hat{u}_h, \hat{z}_h; N_re_s)$
end for

Shape gradient of constraint functions

for all design variables N_re_s **do** ▷ directional derivatives of constraints
  $\nabla g_{r,s} \leftarrow \mathrm{d}\hat{g}(G; N_re_s)$ ▷ derived from equations (3.53)–(3.54)
  $\nabla h_{r,s} \leftarrow \mathrm{d}\hat{h}(G; N_re_s)$
end for

Sobolev smoothing ▷ **if needed**

for all design variables N_ke_l **do** ▷ assemble system matrix
  for all design variables N_re_s **do**
    $R_{klrs} \leftarrow (N_ke_l, N_re_s)$ with scalar product $(\cdot,\cdot)$ of Hilbert space H of wanted regularity, see Section 5.2
  end for
end for
Solve $Ry = \nabla J$, $\nabla J \leftarrow y$
Solve $Ry = \nabla g$, $\nabla g \leftarrow y$
Solve $Ry = \nabla h$, $\nabla h \leftarrow y$

6.1.3 Computation of new geometry

Here, we give the procedure for updating a geometry in Algorithm 1. The descent direction s depends on what the (black box) optimization routine returns on the basis of the gradients ∇J, ∇h and ∇g but it is up to the user how to update the geometry with it and possibly to move the mesh. We summarize the whole procedure in Algorithm 4. As demonstrated with Example 4.5, the update rules differ for NURBS with variable weights in the *discretize first* and *optimize first* ansatz; for a comparison with Algorithm 4 line 6 and onwards, we state the update rule of *discretize first* in the following code snippet

$X_i \leftarrow X_i + (s_{i,1}, \dots, s_{i,d})^\mathsf{T}$
$\omega_i \leftarrow \omega_i + s_{i,d+1}$.

Hence, for $S_i := (s_{i,1}, \dots, s_{i,d})^\mathsf{T}$ and $W_i := s_{i,d+1}$ an update rule with homogeneous coordinates results in a geometry

$$G_{new} = \sum_{i=1}^{n} (X_i \omega_i + S_i) \frac{N_i}{\sum_{j=1}^{n} (\omega_j + W_j) N_j} \tag{6.2}$$

as opposed to the new geometry from the discretize first approach

$$G_{new} = \sum_{i=1}^{n} (X_i + S_i)(\omega_i + W_i) \frac{N_i}{\sum_{j=1}^{n} (\omega_j + W_j) N_j} \ . \tag{6.3}$$

We summarize the update of a geometry in Algorithm 4.

Algorithm 4 UPDATE(Ω, s)

Require: stiffness matrix A of mesh moving problem
Require: geometry function G of initial domain Ω_0
Require: descent direction s from black-box gradient descent optimizer
1: apply mesh movement function $y = \Phi(s)$ to propagate the change also to the inside control points, $s \leftarrow y$
2: **if** G is a B-spline geometry or a NURBS geometry with fixed weights **then**
3: **for all** geometry basis functions N_i **do**
4: $X_i \leftarrow X_i + (s_{i,1}, \dots, s_{i,d})^\mathsf{T}$
5: **end for**
6: **else** G is a NURBS geometry with variable weights
7: **for all** geometry basis functions N_i **do**
8: $X_i \leftarrow X_i + (s_{i,1}, \dots, s_{i,d})^\mathsf{T} / s_{i,d+1}$
9: $\omega_i \leftarrow \omega_i + s_{i,d+1}$
10: **end for**
11: **end if**

6.1.4 Isogeometric shape optimization loop

In this section we evaluate the isogeometric shape optimization process from two –related– points of view. The first one considers actual software decisions for implementation. The second stays on a more abstract level and reconsiders the whole optimization loop for IGA compared to classical FEM shape optimization.

Implementation

In [Dubois, 2005, in B. Meyer], the benefits of third party libraries are praised as less bug prone due to extensive testing by lots of users, not to mention the time saving. We also profit that way by using software from others which we gratefully acknowledge here.

Most importantly, this work is built upon the isogeometric analysis solvers of A. -V. Vuong. We refer to his book [Vuong, 2012] for all implementation aspects around the PDE solver like element-wise assembly, boundary conditions in isogeometric analysis and also for local adaptive refinement by means of hierarchical B-splines. We extended his work to serve shape optimization problems:

- For the `C++` sparse linear algebra we now rely on the software package `Trilinos`, [Sala et al., 2004], where sparse direct and indirect solvers for linear systems are particularly important to us.
- For this thesis we implemented a `C++` optimization class with interfaces to the optimization packages from [Johnson, 2014] and [Wächter and Biegler, 2006]. Therefore, there are different optimization methods available to us, from which we use SQP and MMA from `NLOPT` and the interior point method in `IPOPT`.
- Some of the applications also make use of the `MATLAB` version of the IGA solver and the optimization routines from [MATLAB, 2012].

Now we want to address practical implementation aspects for the optimization loop. These are often organized by calling subroutines for the cost function and constraint evaluations. In shape optimization, costs, and often the constraints, depend on the solution of a PDE. Thus, we need a communication between an optimizer like MMA or SQP and a PDE solver for the state equation, see Figure 6.1. The calling sequence is controlled by the optimization method. Therefore, our optimization problem class stores the previous computations, like basis function evaluations, for the next iteration on Ω_{k+1}.

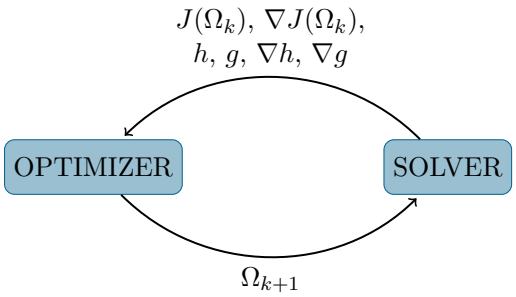


Figure 6.1: Communication between simulation and optimization

The optimization loop revisited

The communication between optimizer and PDE solver from Figure 6.1 is very critical to the success of shape optimization in a numerical sense: The descent direction from the shape gradients needs to fit to the optimization problem; due to discretization this agreement might be destroyed and a successful optimization thwarted. Let us address in the following three objects of discretization,

- geometry approximations,
- shape gradient approximations,
- quadrature approximations.

Geometry approximations Due to the good design properties of B-splines for surfaces and bodies, they are also used in classical finite element shape optimization for boundary representations, for instance in major applications of engineering disciplines like airfoil optimization [Schulz and Borzì, 2012, Chapter 7.2], or in biomedical engineering modeling cardiovascular stents [Clune et al., 2014]. This means for a shape optimization with traditional finite element methods that the domains for the PDE solver are approximated as presented in Section 4.3.1 and depicted for instance in Figure 4.8. It affects the optimization because one looks for an optimal B-spline control point using information from a polygonal domain. This is not consistent, though in the limit of infinitely refined FEM triangulation, the difference should vanish. Since we use in IGA the original geometry this inconsistency falls away in the optimization loop in Figure 6.2.

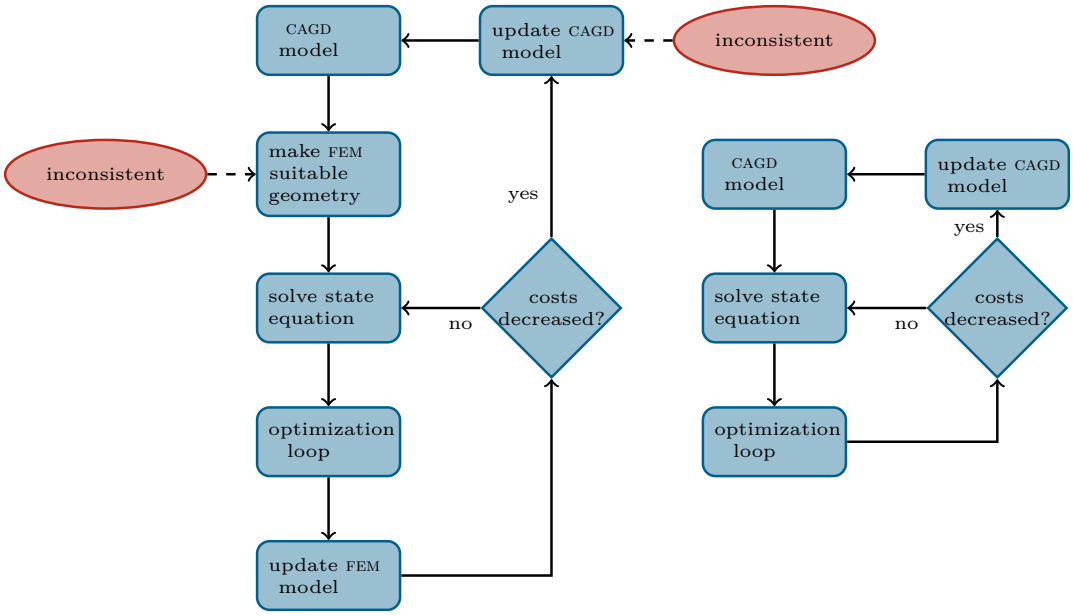


Figure 6.2: Pattern of an optimization loop for FEM (left) and IGA (right): Finite element shape optimization in connection with B-spline geometry models may introduce inconsistencies when approximating the shapes with a triangulation for the PDE solver. This is not the case in IGA which works on the original geometry model for optimization and simulation.

Shape gradient approximations Although we can remove the inconsistent geometry models with IGA the situation might be different for the shape gradient approximation. In FEM shape optimization the split between *optimize first* and *discretize first* is sometimes motivated by the experience that the gradients from the *discretize first* approach seem to fit better to the discrete problem out of consistency reasons (same as in the previous paragraph). In literature, this effect is observed for

instance in [Carnarius et al., 2010] and to our knowledge, it is more pronounced for such time-dependent problems.

With the result from our comparison of the two approaches for IGA in Section 4.4.3, namely that discretization and optimization commutes for linear elliptic second order PDE state equations, we can exclude for our problem class this consideration.

That leaves the last item in our list:

Quadrature approximations Up to now we always assumed that we can solve occurring integrals analytically. Of course, this is not the case and we use Gauß quadrature to approximate the cost functional, constraints, linear and bilinear forms, etc. Again, this is well-known to pose an inconsistency between the shape gradients derived for the mother problem and the discretized problem treated with numerical quadrature. This is not a feature of *optimize first* but can happen to *discretize first* as well. Unfortunately, this situation is not improved by using IGA and we keep it in mind for our applications.

6.2 Shape Optimization Applications

In this section we apply the complete shape optimization process given by Algorithm 1 with the procedures in lines 2–8 to some model problems. These examples illustrate the isogeometric shape optimization process and moreover, expose possible weaknesses, which have to be considered. We treat three optimization problems: maximizing an area, tracking a state and increasing stiffness. The first one involves no PDE, but demonstrates the influence of quadrature errors in optimizing weights and control points simultaneously. The second example tracks a given heat distribution, hence involving a Poisson equation. Its simple form is especially suited to portray the structure of isogeometric transformation and shape sensitivities when PDEs are involved. The third problem is well known in shape optimization and has been treated already with isogeometric analysis. We include it here, for two reasons. Firstly, it is mainly posed in the *discretize first* setting in IGA literature, so here we view it also in the *optimize first* picture. Secondly, in this example we plead for the use of two separate meshes to obtain good results.

6.2.1 Maximizing an area

In this section we review the ancient *isoperimetric problem of Dido* under isogeometric shape optimization. That is, we want to maximize the area of a domain Ω such that the length of its perimeter stays invariant. We choose this example to demonstrate the combination of shape optimization with NURBS and B-splines. It also serves as simple case for an IGA shape gradient derivation. We presented the sensitivity computation and for this example in [Fußeder et al., 2015] for B-splines and for the weight optimization in [Fußeder and Simeon, 2015].

Problem 4 (Area Maximization).
For $\Omega \subset \mathbb{R}^2$ we consider the shape optimization problem

$$\min J(\Omega) := -\int_{\Omega} \mathrm{d}x, \quad s.t. \int_{\partial\Omega} \mathrm{d}\Gamma = P_0. \tag{6.4}$$

The optimal shape then is a circular domain with radius $r = \frac{P_0}{2\pi}$.

To use the *optimize first* approach we first pull the problem back onto the parameter domain, then we obtain the sensitivities for a B-spline case and afterwards for the NURBS case with variable weights as well. We then describe the set-up of the optimization process, present our results and report on the hurdles towards the optimal disk.

Transported problem and sensitivity calculation

Suppose we have a B-spline or NURBS tensor product (like) space with fixed weights $\mathcal{S}^2 = \operatorname{span}\{N_{i,k} = N_i e_k,\, i = 1,\dots,n,\, N_i \in \mathcal{S},\, k = 1,2\}$ and a geometry function $G \in \mathcal{S}^2$ therein: $G = \sum_{i=1}^{n}\sum_{k=1}^{2} X_{i,k} N_i e_k = \sum_{\alpha=(i,k)} X_\alpha N_\alpha$ with control points $X_\alpha \in \mathbb{R}$ as defined on page 60. The transported cost functional reads then on the parameter domain $\hat{\Omega} = (0,1)^2$

$$\min \hat{J}(G) := -\int_{\hat{\Omega}} |\det \mathrm{J}_G| \,\mathrm{d}\hat{\Omega}\,. \tag{6.5}$$

For the length constraint $\hat{h}(G) = 0$ we have from the four parameter boundary faces

$$\hat{h}(G) := \int_0^1 |\mathrm{d}_{\hat{s}} G(\hat{s},0)| + |\mathrm{d}_{\hat{s}} G(\hat{s},1)| + |\mathrm{d}_{\hat{s}} G(0,\hat{s})| + |\mathrm{d}_{\hat{s}} G(1,\hat{s})| \,\mathrm{d}\hat{s} - P_0\,, \tag{6.6}$$

because $\mathrm{J}_G^{-\mathsf{T}}\, \hat{n} \det \mathrm{J}_G = \mathrm{d}_{\hat{s}} G(\gamma(\hat{s}))$, $\gamma(\hat{s}) \in \{(0,\hat{s}),(1,\hat{s}),(\hat{s},0),(\hat{s},1)\}$. We use the transported equations to derive the shape sensitivities for the B-spline case next.

Sensitivities Lemma 3.24 yields the shape derivatives for $\theta = N_\alpha \in \mathcal{S}^2$

$$\mathrm{d}\hat{J}(G; N_\alpha) = -\int_{(0,1)^d} \operatorname{tr}(\mathrm{J}_G^{-1}\, \mathrm{D}N_\alpha) |\det \mathrm{J}_G| \,\mathrm{d}\hat{\Omega}, \tag{6.7}$$

$$\begin{aligned} \mathrm{d}\hat{h}(G; N_\alpha) = \int_0^1 &\frac{\mathrm{d}_{\hat{s}} G(\hat{s},0) \cdot \mathrm{d}_{\hat{s}} N_\alpha(\hat{s},0)}{|\mathrm{d}_{\hat{s}} G(\hat{s},0)|} + \frac{\mathrm{d}_{\hat{s}} G(\hat{s},1) \cdot \mathrm{d}_{\hat{s}} N_\alpha(\hat{s},1)}{|\,\mathrm{d}_{\hat{s}} G(\hat{s},1)|} + \\ &+ \frac{\mathrm{d}_{\hat{s}} G(0,\hat{s}) \cdot \mathrm{d}_{\hat{s}} N_\alpha(0,\hat{s})}{|\,\mathrm{d}_{\hat{s}} G(0,\hat{s})|} + \frac{\mathrm{d}_{\hat{s}} G(1,\hat{s}) \cdot \mathrm{d}_{\hat{s}} N_\alpha(1,\hat{s})}{|\,\mathrm{d}_{\hat{s}} G(1,\hat{s})|} \,\mathrm{d}\hat{s}. \end{aligned} \tag{6.8}$$

To be able to optimize also the weights in case of a NURBS geometry, we consider the rational sensitivities which are obtained in the following.

Rational sensitivities We suppose that the geometry is given in a NURBS space $\mathcal{N}$ with free weights. It is constructed from the non-rational B-splines $\mathcal{S}$ with homogeneous coordinates as defined in Section 4.2.1. Lemma 4.33 yields the shape derivatives in the direction $\theta = N_\alpha \in \mathcal{N}$ for the cost and constraint terms

$$\mathrm{d}\hat{J}(G; N_\alpha) = -\int_{(0,1)^d} \operatorname{tr}(\mathrm{J}_G^{-1}\, \dot{D})) |\det \mathrm{J}_G| \,\mathrm{d}\hat{x}\,, \tag{6.9}$$

$$\mathrm{d}\hat{h}(G;N_\alpha) = \int_{\hat{\Gamma}} \left(|\mathrm{J}_G^{-\mathsf{T}}\,\hat{n}|\,\mathrm{tr}(\mathrm{J}_G^{-1}\,\dot{D}) - \frac{\mathrm{J}_G^{-\mathsf{T}}\,\dot{D}^{\mathsf{T}}\,\mathrm{J}_G^{-\mathsf{T}}}{|\mathrm{J}_G^{-\mathsf{T}}\,\hat{n}|} \right) |\det \mathrm{J}_G|\,\mathrm{d}\hat{x}\,, \tag{6.10}$$

where the shape derivative of the variation in homogeneous representation is

$$\dot{D} = \frac{1}{w^2} \begin{Bmatrix} \begin{pmatrix} 0_2 & -N_\alpha \end{pmatrix} & \alpha = (i, k = 1, 2) \\ \begin{pmatrix} I_2 & -2G \end{pmatrix} & \alpha = (i, 3) \end{Bmatrix} \begin{pmatrix} \mathrm{D}G^w \\ \mathrm{D}w \end{pmatrix} + \frac{1}{w} \begin{pmatrix} I_2 & -G \end{pmatrix} \begin{pmatrix} \mathrm{D}N_\alpha \end{pmatrix}. \tag{6.11}$$

We hand these directional derivatives to the optimization method to get the descent directions. In the following we discuss the results of the numerical optimization with and without weights and how to obtain them.

Optimization methods

We performed the optimization with the SQP algorithm of `MATLAB`'s constrained minimization function `fmincon`, which uses the shape gradient information to update a quasi-Newton approximation of the Hessian of the discrete Lagrangian function $\mathcal{L}$ of the cost functional and the constraint on the control. For this example, $\mathcal{L}(\lambda, G) = \hat{J}(G) + \lambda\hat{h}(G)$ with Lagrange multiplier λ. We proceed first with the non-rational B-spline case and upgrade to rational, weight optimization afterwards.

Results for non-rational optimization

We examined both, the influence of the degree p of our B-spline basis and of the discretization parameter h corresponding to a refinement by knot insertion. Starting with an initial parameterization given in Appendix C, Table C.5 for $p = 1$, the order is increased through repeating the first and last knot, as in Lemma 4.16.

Convergence The convergence plot in Figure 6.3 reveals two things. First, we see that non-rational B-splines behave as one expects: Since higher degree B-splines can approximate (conics) better, they should perform better, i.e. giving smaller errors than lower degree approximations. Second, one anticipates h-convergence for a fixed degree on the same grounds. For $p = 4$ the flat behaviour after two refinements is due to reaching machine precision.

From Figure 6.4 we see that for a higher polynomial degree p, the SQP optimizer in `fmincon` needs significantly less iterations or number of function calls respectively. Intuitively this is what one expects, since the approximation power of B-spline curves to sufficiently smooth functions f goes like $\max(Qf - f) \leq Ch^{p+1}$ for the B-spline interpolation Qf of f, see [De Boor, 2001, Jackson type estimate]. We conclude that a higher degree p of our B-spline basis speeds up convergence.

Mesh movement Choosing an initial B-spline representation such as in Figure 6.5a will yield an irregular parameterization for some step sizes, as indicated in Figure 6.5b. Therefore, to use this configuration for shape optimization, in Example 4 one of the moving mesh algorithms has to be applied. Applying for instance the relative positioning (5.15), again convergence of the optimization algorithm can be observed, compare Figure 6.5c.

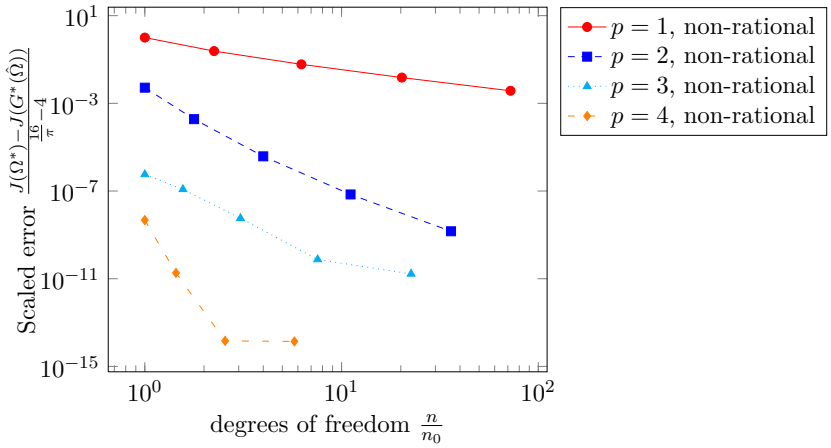


Figure 6.3: Convergence plot for Example 4 where h-refinement corresponds to an increasing number of degrees of freedom n and $\Omega^\star$ is the known optimal domain and $G^\star(\hat{\Omega})$ the numerical optimum by the isogeometric method with *optimize first* scheme. For a better comparison we scale the error by the starting error which is the area of the optimal disk minus area of initial square. Moreover, for each p we also norm the degrees of freedom n by their starting number of degrees of freedom n_0.

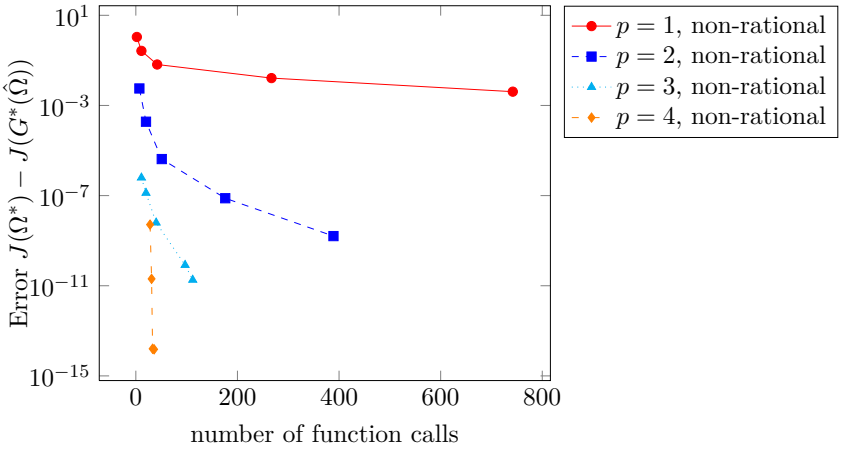
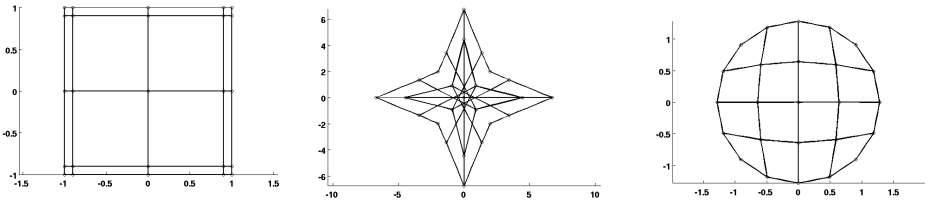


Figure 6.4: Error in terms of computational costs for Example 4, where the costs are expressed as number of function calls by the `MATLAB` optimization method. The number of function calls is closely related to the number of iterations of the optimizer.

Results for rational optimization

A first guess would assign the same behavior to rational B-splines, even expect a much lower error for $p = 2$ already, because the disk has a representation in this space. We use again the knot vectors $(0, 0, 0, 1, 1, 1)$ in both space directions with 3×3 rational B-spline functions and 9 weights, all equal to 1; control points are such that we have the same initial square as above. Since rational B-splines are rational polynomials we have to take quadrature errors into account. We pursue this in the following before starting optimization with NURBS.



(a) Initial mesh **(b)** Mesh tangling **(c)** Relative positioning

Figure 6.5: Example 4 with an arbitrary initial parameterization leads to mesh tangling in some iterations of the SQP optimizer, and therefore to no convergence. However, convergence can be achieved if the moving mesh method (5.15) is applied.

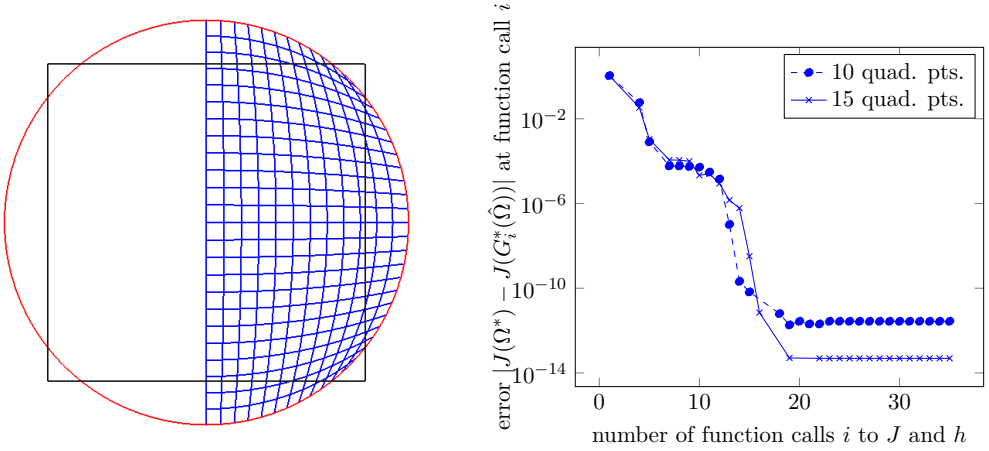
Influence of quadrature errors When picking efficient quadrature rules for the assembly of $\hat{a}(G)$ and $\hat{l}(G)$ in IGA, transformation terms $\det \mathrm{J}_G$ and $\mathrm{J}_G^{-\mathsf{T}}$ are neglected in this search by [Hughes et al., 2010] on the basis that these terms change slowly in comparison to other factors. This is in general valid since the geometry is fixed at a coarse level with degree p_0 and mesh size parameter h_0; for analysis a (several times) p- and h-refined version is used. Hence, G is a polynomial with degree $p \geq p_0$, and is almost constant on elements of sizes $h \ll h_0$. This carries over to the transformation terms. However, this assumption does not hold anymore when the geometry G has the same refinement level or worse, for cost functionals or for constraints on Ω without state, as for instance a volume constraint in the form of $h(\Omega) = \int_\Omega 1 \,\mathrm{d}x$. In IGA, this yields $h(G) = \int_{\hat{\Omega}} |\det \mathrm{J}_G| \,\mathrm{d}\hat{x}$. Using rational B-splines, this simple constraint already exhibits a need for different or higher order quadrature rules. Different quadrature rules could mean Gauß-type rules integrating (some) rational polynomials exactly [Gautschi, 1993]. However, we found it sufficient for our test case to pursue higher order quadrature rules. That is, if we enforce the volume constraint by a tolerance smaller than 10^{-6} in the optimization routine, Table 6.1 shows that we need at least more than 5 quadrature points in each direction to get rid of dominating integration errors.

	Quadrature errors		
#quadrature points	5	9	15
volume	$\approx 8 \times 10^{-6}$	$\approx 3 \times 10^{-11}$	$\approx 1 \times 10^{-15}$
perimeter	$\approx 9 \times 10^{-7}$	$\approx 2 \times 10^{-12}$	$\approx 9 \times 10^{-16}$

Table 6.1: Influence of quadrature errors in Problem 4 for volume and perimeter calculation for disk with radius 1

Convergence For Problem 4 with NURBS of degree $p = 2$ and variable weights the exact disk $\Omega^\star$ is in this NURBS space, so we expect an error $J(\Omega^\star) - J(G^\star(\hat{\Omega}))$ dominated by quadrature errors and optimization parameters. In Figure 6.6 a visually good

result is obtained for the following setting. A Gauß quadrature with 10 quadrature points in each space direction is used. The perimeter constraint is enforce within a tolerance of 10^{-8} and as stopping criteria for the SQP optimizer in the package `NLOPT` [Johnson, 2014] the same tolerance is chosen. In Figure 6.6 no difference between exact and numerical optimum can be detected. In fact, the true error in the cost functional is $|J(\Omega^\star) - J(G^\star(\hat{\Omega}))| \approx 2.1 \times 10^{-10}$ after 34 function calls, i.e. iterations. For an even smaller error we increase the quadrature rule to 15 points in each direction and require that the perimeter may not deviate more than 10^{-12} from $P_0 = 8$ and as stopping tolerance we give 10^{-9}. Then, the error $|J(\Omega^\star) - J(G^\star(\hat{\Omega}))|$ drops to $\approx 8.9 \cdot 10^{-15}$ after 34 function calls. It is unreasonable to expect anything less since from Table 6.1 we learn that the error in the objective is then dominated by quadrature errors. In Table 6.1 are volume and perimeter values for a disk modeled by above NURBS. In particular, we computed these values using Gauß quadrature with different number of quadrature points.



(a) Optimal numerical shape (plotted as evaluation grid from IGA geometry on half disk) is a NURBS approximation of a disk (exact disk plotted as boundary circle)

(b) History of (feasible) SQP optimization steps

Figure 6.6: Result for Problem 4 with the unit square as initial domain

Shape sensitivity validation To validate the shape sensitivities, we have double-checked the shape gradients of $\hat{J}$ and $\hat{h}$ for both, optimization with and without weights in the *optimize first* setting, with `MATLAB` central finite difference test for gradients where $\mathrm{d}\hat{J}(G; N_i e_k) \approx \frac{1}{2\epsilon}\big(J((G + \epsilon N_i e_k)(\hat{\Omega})) - J((G - \epsilon N_i e_k)(\hat{\Omega}))\big)$. These values agree up to the default tolerance in `fmincon` which means a relative error of less than 10^{-6} in each component of the gradient, but typically we even observed a relative error of less than 10^{-13}. Using NURBS the higher quadrature rules have to be taken into account and the default tolerance in the gradient test has to be modified to get a difference unpolluted by quadrature errors.

Comparison of rational and non-rational shape optimization We compare the results from rational and non-rational B-spline optimization and how many design

variables we need in both cases for an error in the cost functional of order 10^{-15}. In case of NURBS with weight optimization we have 9 control points with 8 of them corresponding to basis functions with support on the boundary. Since they have two components, we end up with 16 optimization variables plus 8 weights for each basis, amounting to 24 design variables in total. In case of B-splines without weights Figure 6.3 shows that for the desired error we need B-splines of degree $p = 4$ which have been h-refined at least twice. This leaves us with 64 degrees of freedom of which 32 are nonzero at the boundary. Thus, considering that each control point has two components we have 64 design variables in total.

Although we need less design variables using NURBS we pay for it by using higher quadrature rules. Moreover, from the case $p = 4$ for B-splines we have seen the fast convergence in Figure 6.4. Together with a PDE constraint it might therefore be computationally more efficient to use B-splines of higher degrees even if the optimal shape is expected to be conic. In the following problems we use therefore only B-splines.

6.2.2 Tracking type stationary heat equation

The next example tracks a prescribed state in its objective function. First published in [Kiniger and Vexler, 2013] it is of special interest because the authors provide an *a priori* error analysis for a finite element discretization of this problem. We first discuss its features and then use it to illustrate local refinement with hierarchical B-splines for the simulation.

Problem 5 (Tracking type problem).
The task is to find a state u, for instance a heat distribution, that satisfies a Poisson equation (the PDE Problem 1)

$$u \in \mathcal{V} = H_0^1(\Omega): \begin{cases} -\Delta u &= f \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \partial\Omega \end{cases} \tag{6.12}$$

such that the difference to a prescribed state $\|u-u_d\|^2_{L^2(\Omega)}$ is minimized. This prescribed solution also yields the right hand side $f(x) = -\Delta u_d$ to the state equation. Moreover, the domains Ω are controlled by $q\colon (0,1) \to \mathbb{R}$ which describes the moving boundary Γ, see Figure 6.7 for the set-up. We modify the original cost functional and add a constraint on the domain: We minimize

$$J(\Omega, u) := \frac{1}{2}\|u - u_d\|^2_{L^2(\Omega)}\,, \tag{6.13}$$

where furthermore a volume constraint must not be violated, i.e. we have the control constraint

$$\int_\Omega \mathrm{d}x = V_0 \ \textit{const}. \tag{6.14}$$

We discuss the shape of an optimal domain.

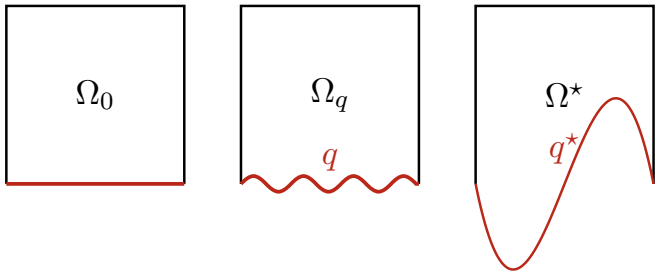


Figure 6.7: Initial domain Ω_0, domains with moving boundary Ω_q and optimal shapes $\Omega^\star$ for the tracking type Problem 5 with prescribed solution from (6.15)

Optimal domains Suppose the moving boundary $\Gamma = \{(x, y) \in \mathbb{R}^2 \colon x \in (0, 1),\ y = q(x)\} \subset \partial\Omega$ is a level set of the prescribed solution $u_d(x, y) = 0$ for $(x, y) \in \Gamma$ then $u = u_d$ almost everywhere and the cost functional J attains its minimum at 0. We track the prescribed state

$$u_d = (x - x^2)(1 - y)(y + 10x(1 - x)(0.5 - x)) \tag{6.15}$$

which leads to an optimal control (optimal function describing the boundary) $q^\star = -10x(1 - x)(0.5 - x)$ depicted in Figure 6.7. Hence, the optimal shape is an element of a B-spline space of degree $p \geq 3$.

To treat the problem with a gradient-based optimization method, we next calculate the isogeometric shape sensitivities.

Shape derivatives

We now compute the shape derivatives obtained with the transformation approach from Chapter 3. For comparison we first state the standard shape derivative for this kind of problem given in [Sokolowski and Zolésio, 1992, p. 124].

The standard shape derivative of the cost functional in the Hadamard structure is given by

$$\mathrm{d}J(\Omega, u; \delta q) = \int_\Gamma (-\nabla u \cdot \nabla z + 0.5(u - u_d)^2)\delta q \cdot n \,\mathrm{d}s\,,$$

with adjoint z satisfying $-\Delta z = u - u_d$ in Ω and $z = 0$ on Γ.

Lemma 6.1. *The isogeometric shape derivative obtained with our transformation approach is given by*

$$\begin{aligned}\mathrm{d}\hat{J}(G, \hat{u}, \hat{z}; \delta\hat{q}) = \int_{\hat{\Omega}} &(\hat{u} - u_d \circ G)(-\nabla u_d \circ G) \cdot \delta\hat{q} + \\ &+ \nabla f \circ G \cdot \delta\hat{q}\hat{z} - \nabla\hat{u}\,\mathrm{J}_G^{-1}\,\mathrm{J}_G^{-\mathsf{T}}\nabla\hat{z}|\det \mathrm{J}_G|\,\mathrm{d}\hat{x}\,.\end{aligned} \tag{6.16}$$

Proof. We show this by considering the transformed terms

$$\hat{J}(G,\hat{u}) = \frac{1}{2}\int_{\hat{\Omega}} (\hat{u} - u_d \circ G)^2 |\det \mathrm{J}_G| \,\mathrm{d}\hat{x}\,, \tag{6.17}$$

$$\hat{l}(G)(\hat{v}) = \int_{\hat{\Omega}} f \circ G\hat{v} |\det \mathrm{J}_G| \,\mathrm{d}\hat{x}\,, \tag{6.18}$$

$$\hat{a}(G)(\hat{u},\hat{v}) = \int_{\hat{\Omega}} \nabla\hat{u} \cdot \mathrm{J}_G^{-1}\, \mathrm{J}_G^{-\intercal}\, \nabla\hat{v} |\det \mathrm{J}_G| \,\mathrm{d}\hat{x}\,, \tag{6.19}$$

which form the Lagrangian

$$\mathcal{L}(G,\hat{u},\hat{z}) = \hat{J}(G,\hat{u}) + \hat{l}(G)(\hat{z}) - \hat{a}(G)(\hat{u},\hat{z})\,. \tag{6.20}$$

In the Lagrange functional the variables are independent and we obtain the directional derivatives by Lemma 3.21

$$\begin{aligned}\hat{J}'_{\hat{q}}(G,\hat{u})(\delta\hat{q}) = \int_{\hat{\Omega}} \Big(&(\hat{u} - u_d \circ G)(-\nabla u_d \circ G \cdot \delta\hat{q}) \\ &+ \frac{1}{2}(\hat{u} - u_d \circ G)^2 \operatorname{tr}(\mathrm{J}_G^{-1}\, \mathrm{J}_{\delta\hat{q}}) \Big) |\det \mathrm{J}_G| \,\mathrm{d}\hat{x}\,,\end{aligned} \tag{6.21}$$

$$\hat{J}'_{\hat{u}}(G,\hat{u})(\delta\hat{u}) = \int_{\hat{\Omega}} (\hat{u} - u_d \circ G)\delta\hat{u} |\det \mathrm{J}_G| \,\mathrm{d}\hat{x}\,, \tag{6.22}$$

$$\hat{l}'_{\hat{q}}(G)(\delta\hat{q},\hat{z}) = \int_{\hat{\Omega}} \Big(\nabla f \circ G \cdot \delta\hat{q}\hat{z} + f \circ G\hat{z} \operatorname{tr}(\mathrm{J}_G^{-1}\, \mathrm{J}_{\delta\hat{q}}) \Big) |\det \mathrm{J}_G| \,\mathrm{d}\hat{x}\,, \tag{6.23}$$

$$\begin{aligned}\hat{a}'_{\hat{q}}(G)(\delta\hat{q},\hat{u},\hat{z}) = \int_{\hat{\Omega}} \Big(&-\nabla\hat{u} \cdot \mathrm{J}_G^{-1}(\mathrm{J}_{\delta\hat{q}}\, \mathrm{J}_G^{-1} + \mathrm{J}_G^{-\intercal}\, \mathrm{J}_{\delta\hat{q}}^{\intercal})\, \mathrm{J}_G^{-\intercal}\, \delta\hat{q}\, \nabla\hat{z} \\ &+ \nabla\hat{u} \cdot \mathrm{J}_G^{-1}\, \mathrm{J}_G^{-\intercal}\, \nabla\hat{z} \operatorname{tr}(\mathrm{J}_G^{-1}\, \mathrm{J}_{\delta\hat{q}}) \Big) |\det \mathrm{J}_G| \,\mathrm{d}\hat{x}\,.\end{aligned} \tag{6.24}$$

The adjoint $\hat{z} \in \hat{\mathcal{V}}$ is given as solution of the variational equation $\mathrm{d}_{\hat{u}}\mathcal{L}(G,\hat{u},\hat{z};\delta\hat{u}) = 0$ for all $\delta\hat{u} \in \hat{\mathcal{V}}$. A comparison with the classical shape derivative given above reveals that

$$\hat{J}'_{\hat{u}}(G,\hat{u})(\delta\hat{u}) = \hat{a}(G)(\delta\hat{u},\hat{z}) \quad \forall \delta\hat{u} \in H_0^1(\hat{\Omega})$$

is just its transformed weak form. □

In the following, we discuss the set-up of our isogeometric shape optimization and the use of adaptive refinement for the simulation.

Results

We test with Problem 5 adaptively refined simulation meshes with hierarchical B-spline introduced in Section 5.5. We implemented this scenario using the `MATLAB` optimizer `fmincon` together with its finite difference options to compute the gradients of J. For the volume constraint we use the analytical shape gradients from the *optimize first* method. If we view the shape in the xy-plane, the design variables are the y-components of the control points at the moving boundary, or equivalently, in the *optimize first* setting, the B-spline basis functions N_i with support at the boundary in the second space direction, $N_i e_2$. The mesh is adjusted by relative positioning, compare (5.15), since we only move up and down. The optimizer converged to the shape in Figure 6.8 which indicates that shape optimization behaves well with hierarchically refined simulation meshes. The refinement is steered with a multilevel error indicator described in the author's conference paper [Vuong and Fußeder, 2013].

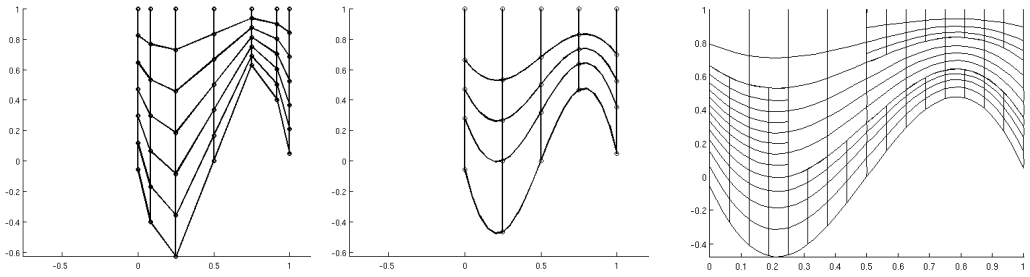


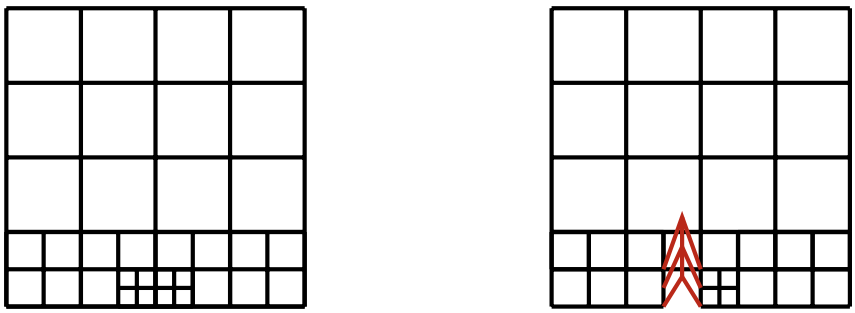
Figure 6.8: Optimization results for Problem 5 with B-splines of degree $p = 2$: numerical optimal parameterization (left), optimal geometry (middle) and adaptively refined simulation mesh in an intermediate step (right)

Hierarchical B-splines suggest further directions for investigation.

Outlook on adaptive refinement Our example demonstrates isogeometric shape optimization can be extended to adaptive refinement in the simulation process. This raises the question if we can also refine the optimization mesh for instance using a geometry basis which is resolved finer at the moving boundary as in Figure 6.9a. From [Giannelli et al., 2012] we know that hierarchical B-splines or their extension to truncated hierarchical B-splines can capture local features of a 2-dimensional manifold in $\mathbb{R}^3$. This suggests that they are ideal for optimizing surface changes orthogonal to the tangent space. For our problem at hand locally refined geometry meshes are more complicated because we have to devise a new mesh movement algorithm: If there is a change on the locally refined region, meshes from different levels may overlap as in Figure 6.9b. A criterion for moving hierarchical meshes for optimization, taking the polynomial degrees into account, is an open question and seems intricate.

We return to the Problem 5 and consider the set-up in [Kiniger and Vexler, 2013].

Outlook on error estimation In its original form in [Kiniger and Vexler, 2013] Problem 5 has a cost functional with a Tikhonov type regularization term instead of



(a) Initial mesh, locally refined on the boundary **(b)** Overlapping levels of refinement

Figure 6.9: Locally refined geometry mesh with 3 different levels: Moving level control points from level 3 causes an overlap with the two coarser meshes. The coarser control polygons cannot be adjusted, or not adjusted easily, to yield a feasible mesh.

our volume constraint,

$$J_{orig}(\Omega, u) := \int_{\Omega} \frac{1}{2}(u - u_d)^2 \,\mathrm{d}x + \frac{\alpha}{2}\|q''\|^2_{L^2(0,1)} \,. \tag{6.25}$$

and a prescribed solution

$$u_{d,orig} := \sin 2\pi x(1-y)(y + \frac{1}{2}\sin 2\pi x) \,. \tag{6.26}$$

Regularization terms are a frequently used tool in optimal control (Remark 2.23) and the authors showed, using the regularization and a transformation approach, the following *a priori* error estimate: Let h_{sim} be the discretization parameter of the finite element method and h_{geo} that of the control, then the error between the numerical optimal control $q^\star_{h_{sim},h_{geo}}$ an an analytical, local solution is estimated as

$$\|q^\star - q^\star_{h_{sim},h_{geo}}\|_{H^2(\Gamma)} \leq c(h^2_{sim} + h^2_{geo}) \,. \tag{6.27}$$

The predicted rates were realized with cubic Hermite finite elements (Figure 4.9b).

For us, this result is significant because we would like to match the estimate (6.27) for IGA theoretically, and possibly extend it to consider the higher continuity and polynomial degree of B-splines (more than cubic). Moreover, due to the presence of Dirichlet boundary conditions the Lagrangian function has to be modified for that (Remark 2.32 for optimal control). Our abstract framework can serve as a starting point for these theoretical investigations.

Practically, the result in (6.27) gives a means to compare IGA and FEM shape optimization. For a successful implementation with IGA there are two challenges: Transported sine functions occur due to $u_{d,orig} \circ G$ in the shape derivatives (6.16) of J_{orig} and in the adjoint state equation, which is integrated in IGA by static Gauß quadrature. Therefore, one has to anticipate quadrature errors. (That is the reason we changed our cost functional to the polynomial version.) The second challenge is the cost control term $\frac{\alpha}{2}\|q''\|^2_{L^2(0,1)}$: For an $\alpha > 0$ that ensures convergence, it is so prohibitive that the optimal domain is again the initial square, [Kiniger and Vexler, 2013]. Therefore, the regularization term could be so influential that a convergence for an IGA shape optimization is not meaningful, if there are no theoretical rates for comparison as in (6.27).

6.2.3 Compliance minimization in linear elasticity

One of the classical shape optimization problems is minimizing the compliance in a plane stress linear elasticity setting, PDE Problem 3,

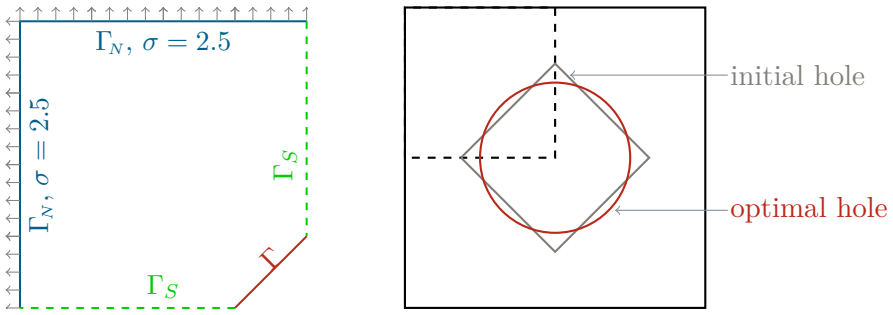
$$\begin{cases} -\operatorname{div}\sigma(u) &= 0 & \text{in } \Omega \\ u &= 0 & \text{on } \Gamma_D \\ \sigma(u)\cdot n &= g_N & \text{on } \Gamma_N \end{cases} \tag{6.28}$$

with strain $\varepsilon(u) = \frac{1}{2}(\nabla u + \nabla u^{\mathsf{T}})$ and stress $\sigma(u) = 2\mu\varepsilon(u) + \lambda(\nabla\cdot u)id$.

A specific problem is the plate with circular hole: Given a plate Ω with hole, one tries to find the shape of the hole such that the deformation work through external forces, the compliance, is minimized, hence the stiffness of the plate increases. This problem has already been been treated by means of IGA in [Qian, 2010, Wall et al., 2008] for the *discretize first* ansatz and in [Blanchard et al., 2013] also for the *optimize first* method.

Problem 6 (Compliance Minimization).
We seek to minimize the compliance $\min J(\Omega, u) := \int_{\Gamma_N} g\cdot u\,\mathrm{d}\Gamma$ *where* u *solves* (6.28) *under an additional volume constraint on the control,* $\int_\Omega \mathrm{d}\Omega = V_0 = const.$ *Because the problem is symmetric, we use only a quarter of the plate with symmetric boundary conditions on* Γ_S.

Existence of optimal shapes for this example has been shown, for instance, in [Haslinger and Mäkinen, 2003]. Typically, one expects a circular hole as optimal solution from calculations in [Timoshenko and Goodier, 1951, p. 88 ff] if the ratio of the length of the hole side to the length of the plate sides is small than $1:4$. The problem set-up of [Wall et al., 2008, Qian, 2010, Blanchard et al., 2013] is displayed in Figure 6.10. We used a Poisson ratio $\nu = 0.3$ and $E = 10^5$ which are comparable to steel, cf. in Table C.1 in the Appendix C.



(a) Boundary conditions **(b)** Initial and optimal shapes

Figure 6.10: Configuration of state equation of Problem 6, as well as initial and optimal shapes

Standard shape gradients The shape gradient in Hadamard structure is given by directional derivatives

$$\mathrm{d}J(\Omega; h) = \int_{\Gamma_N} \left[2\mu|\varepsilon(u)|^2 + \lambda|\operatorname{div} u|^2\right] h\cdot n\,\mathrm{d}\Gamma \tag{6.29}$$

since the compliance minimization is self-adjoint, $u = z$.

Isogeometric shape gradients For isogeometric analysis we change the basis to obtain a weak formulation on function spaces over the parameter domain like (4.44). Problem 6 thus has the bilinear form

$$\hat{a}(G)(N_{\alpha_1}, N_{\alpha_2}) = \int_{\hat{\Omega}} \hat{\underline{\varepsilon}}_G(N_{\alpha_1}) C^{\mathsf{T}} \hat{\underline{\varepsilon}}_G(N_{\alpha_2}) |\det \mathrm{J}_G| \,\mathrm{d}\hat{x} \tag{6.30}$$

in isogeometric analysis where the strain in Voigt notation

$$\hat{\underline{\varepsilon}}_G(\hat{v}) = \underline{\varepsilon}(v) \circ G = \begin{pmatrix} \mathrm{d}_{x_0} v_0 \\ \mathrm{d}_{x_1} v_1 \\ \mathrm{d}_{x_1} v_0 + \mathrm{d}_{x_0} v_1. \end{pmatrix} \circ G = \begin{pmatrix} (\mathrm{J}_G^{-\mathsf{T}} \nabla v)_{1,1} \\ (\mathrm{J}_G^{-\mathsf{T}} \nabla v)_{2,2} \\ (\mathrm{J}_G^{-\mathsf{T}} \nabla v)_{1,2} + (\mathrm{J}_G^{-\mathsf{T}} \nabla v)_{2,1} \end{pmatrix}. \tag{6.31}$$

Assume constant Neumann boundary conditions g_N, then we have for an isogeometric shape gradient in domain representation

$$\mathrm{d}\hat{J}(G; N_\alpha) = \int_{\hat{\Omega}} \hat{\underline{\varepsilon}}_G(\hat{u})^{\mathsf{T}} C \hat{\underline{\varepsilon}}_G(\hat{u}) \operatorname{tr}(\mathrm{J}_G^{-1} \,\mathrm{D} N_\alpha) |\det \mathrm{J}_G| \,\mathrm{d}\hat{x} + \tag{6.32}$$

$$- \int_{\hat{\Omega}} (\dot{\hat{\underline{\varepsilon}}}_G(\hat{u})^{\mathsf{T}} C \hat{\underline{\varepsilon}}_G(\hat{u}) + \hat{\underline{\varepsilon}}_G(\hat{u})^{\mathsf{T}} C \dot{\hat{\underline{\varepsilon}}}_G(\hat{u})) |\det \mathrm{J}_G| \,\mathrm{d}\hat{x} \tag{6.33}$$

with the strain shape sensitivities in the direction of N_α

$$\dot{\hat{\underline{\varepsilon}}}_G(\hat{u}) = \mathrm{d}_t \varepsilon(u) \circ (G + tN_\alpha)|_{t=0} = \begin{pmatrix} (\dot{\mathrm{J}_G^{-\mathsf{T}}} \nabla v)_{1,1} \\ (\dot{\mathrm{J}_G^{-\mathsf{T}}} \nabla v)_{2,2} \\ (\dot{\mathrm{J}_G^{-\mathsf{T}}} \nabla v)_{1,2} + (\dot{\mathrm{J}_G^{-\mathsf{T}}} \nabla v)_{2,1} \end{pmatrix}, \tag{6.34}$$

where

$$\dot{\mathrm{J}_G^{-\mathsf{T}}} = \mathrm{d}_t \, \mathrm{J}_{G+tN_\alpha}^{-\mathsf{T}} = -\,\mathrm{J}_G^{-\mathsf{T}} (\mathrm{d}_t \, \mathrm{J}_{G+tN_\alpha}|_{t=0})^{\mathsf{T}} \, \mathrm{J}_G^{-\mathsf{T}} = -(\mathrm{J}_G^{-1} \,\mathrm{D} N_\alpha \, \mathrm{J}_G^{-1})^{\mathsf{T}} \tag{6.35}$$

from the implicit function theorem.

Mesh movement In this example, the mesh movement of inner control points is realized by means of the linear elasticity operator instead of a Laplacian or relative positioning. Since we have calculated the stiffness matrix for solving the linear elasticity equation anyway, we can recycle it for the mesh movement: the shape gradient yields the right hand side, and also different boundary conditions must be applied, but the bulk of expense from assembling the stiffness matrix has been paid before.

Results

In Section 5.4 we argued that it is possible to use two representations, one for solving the state equation and one for optimization. However, here we claim that it is even necessary: for the compliance, an error in the state u directly influences the cost functional. Suppose u is the exact solution on a domain Ω and u_h its numerical

approximation and $\epsilon = u - u_h$ the error. Then the influence of the error with respect to the state in the objective reads

$$J(\Omega, u) - J(\Omega, u_h) = \int_{\Gamma_N} g_N \cdot u \, \mathrm{d}x - \int_{\Gamma_N} g_N \cdot u_h \, \mathrm{d}x = \int_{\Gamma_N} g_N \epsilon, \tag{6.36}$$

which also has an impact on the shape gradient of J. See Figure 6.11 for a behavior of the compliance for the optimal domain under h-refinement.

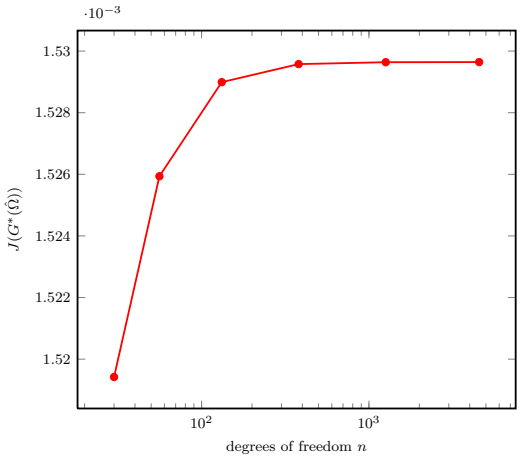
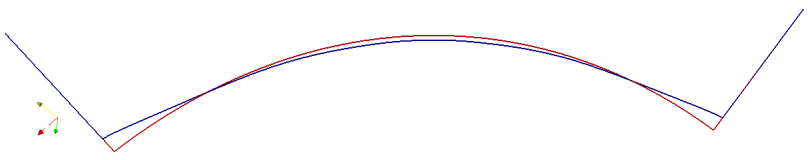
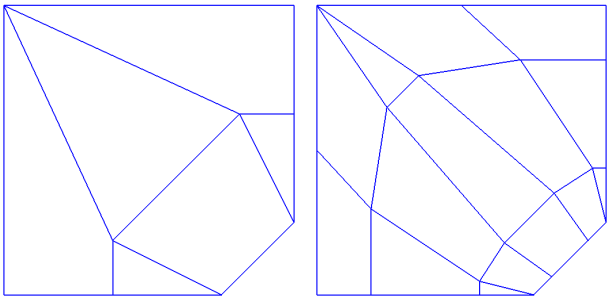


Figure 6.11: Influence of the numerical error in the PDE on the objective function for Problem 6: As the error in the state $u - u_h$ decreases, the compliance, too, is approximated better.

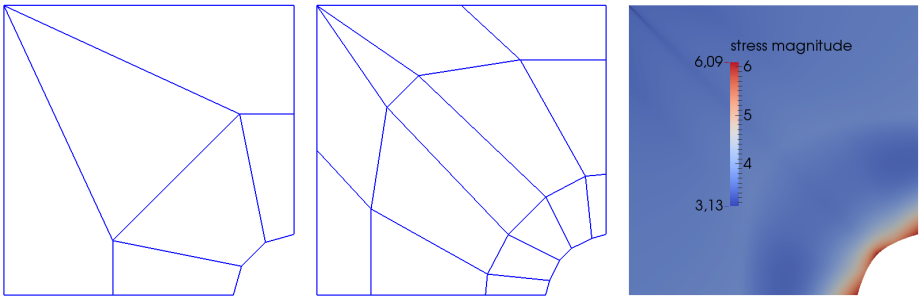
For the geometry data of the initial parameterizations we used a one p-refined version of the data in the Appendix, Table C.4. In Figure 6.12b we show the initial parameterizations and in Figure 6.12c the results for an optimization run, where we use a coarse optimization and a once h-refined analysis mesh. Although the optimizer converges towards a minimum there still is a certain difference to the circular hole as is illustrated in Figure 6.12a. As long as the mesh for solving the state equation is not fine enough, the optimizer will not be able to further enhance the optimal shape. On the other hand, starting the optimization with an already refined geometry is more costly than it needs to be. To support this point, we treat now the scenario of the same coarse optimization mesh and a simulation mesh which is 3 times h-refined, see Figure 6.13a for initial parameterizations. Figure 6.13b shows that indeed with this refined analysis mesh the optimizer is now able to resolve a better approximation to a quarter circle. Note, that it can never be exactly circular, since we use only B-splines here. For both optimization runs we used the SQP solver from the optimization library [Johnson, 2014]. The volume constraint was realized as equality constraint enforced with a tolerance of 10^{-6} and the relative stopping criteria was also set to this tolerance.



(a) Deviation from the exact optimal shape –a circular hole– with coarse analysis mesh

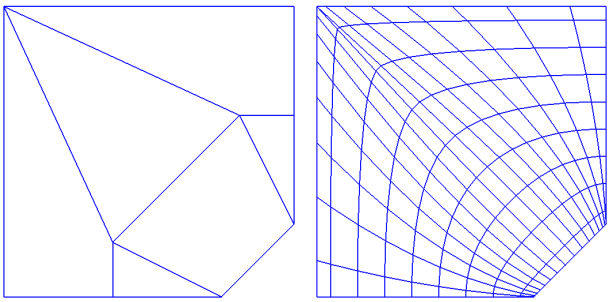


(b) Initial parameterizations of optimization mesh (left) and analysis mesh(right)

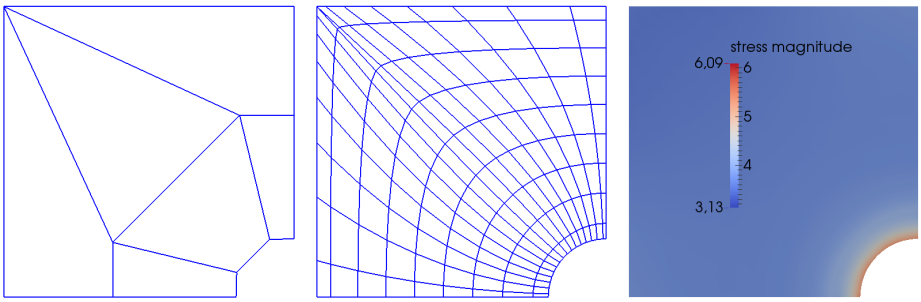


(c) Final parameterizations (left: optimization, middle: analysis) and stress analysis

Figure 6.12: Problem 6 with a once refined analysis mesh



(a) Initial parameterizations of optimization mesh (left) and analysis mesh(right)



(b) Final parameterizations (left: optimization, middle: analysis) and stress analysis

Figure 6.13: Problem 6 with a 3-times refined analysis mesh

7 Conclusion

Finally, we wrap this thesis up with a summary of the main points and a conclusion on the use of isogeometric analysis for shape optimization.

We have provided a theoretical framework for shape optimization with isogeometric analysis based on the infinite-dimensional shape optimization problem. In particular, we used a Lagrange formalism to obtain optimality conditions and shape gradients for gradient-based optimization methods. To absorb isogeometric analysis we used a transformation approach to restate the shape optimization problem over a parameter domain and search for optimal transformations instead of optimal shapes. In this way, isogeometric shape optimization is the discretization of the state space (the function space for the PDE) and the control space (the function space for the shapes) by B-splines or NURBS. We pursued this abstract approach for two reasons.

In the first place, only from the infinite-dimensional problem can existence of optimal solutions be shown. In this light, using our transformation approach and interpreting isogeometric shape optimization as discretization scheme shows the influence of errors from spline approximations more clearly. Therefore, this approach can be a starting point for a convergence study with IGA in future.

Secondly, this *optimize first–discretize then* method offers conceptual insight to shape optimization for instance from optimal control. Thus from this general viewpoint, also techniques in other approaches can be understood. This is especially true for our framework, in which the *discretize first–optimize then* way in IGA is only a different interpretation of derivatives than in *optimize first–discretize then*: We have shown that for linear elliptic partial differential equations of second order the discrete systems of *discretize first* and *optimize first* are equivalent for a discretization with B-splines and fixed weight NURBS. Hence, the abstract framework provides a unified approach to the combination of isogeometric analysis and shape optimization.

Our scheme also covers the discretization by NURBS with variable weights for which we have to use homogeneous coordinates to obtain a feasible set-up. However, then the discrete systems from *optimize first* and *discretize first* differ and have different domain update rules. In order to separate the errors of state and control we showed that using different B-splines/NURBS is supported as well. This therefore also embraces local refinement for the simulation.

From this framework we distilled a step-by-step shape optimization routine for IGA and showed the practical realization of *optimize first* by some examples. For a successful shape optimization process, the interplay of simulation and optimization is crucial. We therefore benefit from this practical framework because it exposes the tuning parameters of the shape optimization procedure: In order for the (black-box) optimization program to converge, the quality of numerical computations must meet its stopping tolerance.

Our example implies that this means for optimizing weights of a NURBS geometry that quadrature errors have to be taken seriously. Being rational polynomials, the

integration error of NURBS in a volume constraint or cost functional is more pronounced than for B-splines and has to be pushed below the tolerance of the optimization program. We have demonstrated that increasing the number of Gauß points deals with this issue successfully in our example.

Another error source is the discretization error of the PDE which has a dramatic effect on the outcome of a shape optimization. In order to keep the number of degrees of freedom for optimization as low as possible while simultaneously achieving better simulation results, we separated the state and control discretization in the compliance minimization example. In a second step, we explored the use of adaptively refined meshes for the PDE in a tracking type example. Based on our framework, a thorough error analysis is possible and can be improved by comparing the results with the ones from traditional methods like FEM.

To sum up, isogeometric analysis is very suitable for shape optimization. Since, however, the optimization process and the interplay between simulation and optimization solver is very delicate, the theoretical and practical framework is useful to understand the individual steps within the whole. In the end, we conclude that there is a huge potential of isogeometric analysis for shape optimization due to the mutual discretization of geometries and PDE solutions with B-splines or NURBS.

A Useful Identities

A.1 Vector Calculus

A.1.1 Divergence operator

The divergence operator div is defined for a vector field $v \in \mathbb{R}^d$ as $\operatorname{div} v = \sum_{i=1}^{d} \partial_i v_i$. For a scalar field $\phi\colon \mathbb{R}^d \to \mathbb{R}$ we have the following useful identity

$$\operatorname{div}(\phi v) = v \cdot \nabla\phi + \phi \operatorname{div} v = \nabla\phi \cdot v + \phi \operatorname{tr} \mathrm{J}_v \,. \tag{A.1}$$

A.1.2 Determinants

Let $A = A(t) \in \mathbb{R}^{d\times d}$ define an invertible matrix where each of its entries is a differentiable function of t. We denote $A' = (\mathrm{d}_t A_{ij})_{ij}$ the derivative of A at t. Then the derivative of its determinant can be obtained with the Leibniz formula which eventually leads to

$$\mathrm{d}_t \det A = \det A \operatorname{tr}(A^{-1}A') \,. \tag{A.2}$$

A.1.3 Chain rule

Let Ω and $\hat{\Omega} \subset \mathbb{R}^d$. We build the composition of a function $u\colon \Omega \to \mathbb{R}$, $x \mapsto u(x)$ with a geometry function

$$G\colon \hat{\Omega} \to \Omega, \quad \xi \mapsto G(\xi) = \begin{pmatrix} G_1(\xi) & \cdots & G_d(\xi) \end{pmatrix}^\intercal$$

which we mark with a hat $\hat{u} \coloneqq u \circ G$, $\hat{u}\colon \hat{\Omega} \to \mathbb{R}$.

Chain rules: 1. derivative

For any composite function $\hat{u} = u \circ G$ the chain rule yields

$$\mathrm{D}(u \circ G) = ((\mathrm{D}u) \circ G)\, \mathrm{D}G = \mathrm{J}_u(G)\, \mathrm{J}_G = \mathrm{J}_{u\circ G} \,. \tag{A.3}$$

For scalar functions u and $\hat{u}$ this yields $\nabla\hat{u} = (\mathrm{D}\hat{u})^\intercal = \mathrm{J}_G^\intercal (\nabla u) \circ G$.

Chain rules: 2. derivative

The second partial derivatives of a composite function are collected in the Hessian which yields

$$\begin{aligned} \mathrm{H}(\hat{u}) &= \mathrm{D}(\mathrm{D}\hat{u})^\intercal = \mathrm{D}\left(\mathrm{J}_G^\intercal (\nabla u) \circ G\right) = \mathrm{D}(\mathrm{J}_G^\intercal) \cdot (\nabla u) \circ G + \mathrm{J}_G^\intercal \cdot \mathrm{D}((\nabla u) \circ G) = \\ &= \mathrm{H}(G) \cdot (\nabla u) \circ G + \mathrm{J}_G^\intercal\, \mathrm{H}(u) \circ G\, \mathrm{J}_G \,. \end{aligned} \tag{A.4}$$

Important notation for isogeometric analysis

More important in IGA, though, is firstly

$$(\mathrm{D}u) \circ G = \mathrm{J}_u(G) = (\mathrm{D}\hat{u})\,\mathrm{J}_G^{-1} \quad \Leftrightarrow \quad (\nabla u) \circ G = \mathrm{J}_G^{-\mathsf{T}}\,\nabla\hat{u}. \tag{A.5}$$

Secondly, $\mathrm{H}(u) \circ G = \mathrm{J}_G^{-\mathsf{T}}\big(\mathrm{H}(\hat{u}) - \mathrm{H}(G)\cdot(\nabla u)\circ G\big)\,\mathrm{J}_G^{-1}$ which is

$$\overset{(A.5)}{=} \mathrm{J}_G^{-\mathsf{T}}\big(\mathrm{H}(\hat{u}) - \mathrm{H}(G)\cdot \mathrm{J}_G^{-\mathsf{T}}\,\nabla\hat{u}\big)\,\mathrm{J}_G^{-1}\,. \tag{A.6}$$

Hence,

$$\Delta\, u(G) \equiv (\Delta\, u)\circ G = \mathrm{tr}(\mathrm{H}(u)\circ G) = \mathrm{tr}\Big[\mathrm{J}_G^{-\mathsf{T}}\big(\mathrm{H}(\hat{u}) - \mathrm{H}(G)\cdot \mathrm{J}_G^{-\mathsf{T}}\,\nabla\hat{u}\big)\,\mathrm{J}_G^{-1}\Big]. \tag{A.7}$$

B Analytical Background

B.1 Compact Subsets

The following definitions can be found in [Alt, 2012, Ch. 2].

Definition B.1. A subset A of a metric space (X, d) is called *compact*

(i) if A is *sequentially compact*, that is, if each sequence in A has a convergent subsequence with limit in A, or equivalently,

(ii) if (A, d) is complete and A *precompact*, that is, for all $\varepsilon > 0$ exists a finite cover of open ε-balls.

Definition B.2. A family of continuous functions $A \subset \mathcal{C}(S; Y)$ from $S \subset \mathbb{R}^d$ compact to a finite-dimensional space (Y, d) is *equicontinuous* if

(i) $\sup_{f \in A} \sup_{x \in S} |f(x)| < \infty$

(ii) $\sup_{f \in A} |f(x) - f(y)| \to 0$ for $x, y \in S \colon |x - y| \to 0$.

B.2 Linear Functional Analysis

B.2.1 Banach and Hilbert spaces

(i) A *Banach space* is a normed space X which is complete, i.e. all Cauchy sequences converge in X.

(ii) Let X denote a real vector space with norm $\|\cdot\|_X$. It is called *Hilbert space* if it is complete and has an inner product $(\cdot, \cdot) \colon X \times X \to \mathbb{R}$.

B.2.2 Linear operators

The definitions and properties of linear operators can be found in any functional analysis textbook, we follow [Ambrosetti and Prodi, 1993] if not stated otherwise.

(i) X, Y normed real vector spaces with norms $\|\cdot\|_X$ and $\|\cdot\|_Y$. A mapping $A \colon X \to Y$ is called *linear* if it satisfies $A(\lambda a + \mu b) = \lambda Aa + \mu Ab \ \forall a, b \in X, \lambda, \mu \in \mathbb{R}$.

(ii) $\mathcal{L}(X, Y)$ denotes the space of all such linear operators that are *bounded* in the sense that

$$\|A\|_{X,Y} := \sup_{\|x\|_X = 1} \|Ax\|_Y < \infty.$$

Here and in the following we often write Ax instead of $A(x)$.

(iii) A is *continuous* if $x_n \to x$ for $n \to \infty$ implies $Ax_n \to Ax$ for $n \to \infty$.

(iv) For linear operators, the *compatibility property* $\|Ax\|_Y \leq \|A\|_{X,Y}\|x\|_X$ holds which implies the equivalence of boundedness and continuity.

(v) On a Banach space X, we call $f \in \mathcal{L}(X, \mathbb{R})$ a *bounded linear functional* on X and

(vi) $X^* := \mathcal{L}(X, \mathbb{R})$ *dual space* of X.

(vii) The *dual pairing* of X^* and X is given by the notation $\langle \cdot, \cdot \rangle_{X^*,X}$, where

$$\langle f, x \rangle_{X^*,X} := f(x).$$

(viii) A Banach space X is *reflexive* if $X \simeq X^{**}$.

(ix) Let X, Y be Banach spaces. Then for an operator $A \in \mathcal{L}(X, Y)$ the *dual operator* $A^* \in \mathcal{L}(Y^*, X^*)$ is defined by

$$\langle A^* y, x \rangle_{X^*,X} = \langle y, Ax \rangle_{Y^*,Y} \quad \forall y \in Y^*,\, x \in X$$

and $\langle A^* y, x \rangle_{X^*,X} =: \langle x, A^* y \rangle_{X,X^*}$.

Theorem B.3 (Riesz representation theorem).
Let X be a real Hilbert space and $f \in X^*$. Then there is a unique $y \in X$ for which

$$\langle f, x \rangle_{X^*,X} = (y, x)\, \forall x \in X\,.$$

Proof. We have stated the theorem according to [Atkinson and Han, 2001, Thm. 2.5.8, p. 82], the proof can be found there, too. □

Lemma B.4. *A linear operator $A\colon X \to Y$ between normed spaces X, Y is continuous on $X \Leftrightarrow$ if it is bounded on X.*

Lemma B.5. *Between a linear continuous operator $A\colon X \to X^*$ and continuous bilinear forms $a\colon X \times X \to \mathbb{R}$ on a real Banach space X exists a one-to-one correspondence given by the dual pairing*

$$\langle Au, v \rangle_{X^*,X} = a(u, v) \quad \forall u,\, v \in X\,, \tag{B.1}$$

e.g. [Atkinson and Han, 2001, Thm. 8.3.1 p. 334].

B.2.3 Compactness in infinite-dimensional spaces

In $\mathbb{R}^d$ the theorem of Bolzano-Weierstraß assures that each bounded sequence has a convergent subsequence. With that, one typically shows for finite-dimensional optimization problems $\min J(x)$ for $x \in A \subset \mathbb{R}^d$ that there is a minimizing sequence $x \to x^\star$ converging to an element in the feasible set A. However, this does not hold for infinite spaces, instead, one considers only weak convergence. More precisely, we compare the finite-dimensional argument to the one in infinite spaces.

Compactness in finite dimensions In finite dimensions we can extract from any sequence in a compact space X a subsequence convergent to an element in X. A characterization of compactness for $X = \mathbb{R}^N$, $N < \infty$, is that it is compact if it is bounded and closed, due to Heine-Borel.

Compactness in infinite dimensions In infinite spaces that is not enough anymore. Instead, consider a reflexive Banach space X (see Section B.2.2, (vi)). If $A \subset X$ is bounded, closed, and convex then it is *weakly sequentially compact*. That means, that from any sequence in A we can extract a subsequence that *weakly converges* to an element in A, see Definition B.6.

B.2.4 Weak Convergence

Definition B.6 (Weak convergence)**.** Let X denote a Banach space and $\{u_k\} \subset X$ a sequence. The sequence *converges weakly* to $u \in X$

$$u_k \rightharpoonup u \text{ in } X \quad \text{if} \quad f(u_k) \to f(u) \, \forall f \in X^* \, .$$

Lemma B.7. *Each weakly convergent sequence is bounded.*

Lemma B.8. *A closed convex subset of $C \subset X$ is weakly sequentially closed. I.e. if $x_k \rightharpoonup x$ for $x_k \in C$ then the limit is also an element in C.*

Lemma B.9. *Let X denote a reflexive Banach space ($X \simeq X^{**}$) and $u_k \subset X$ a bounded sequence. Then there exists a weakly convergent subsequence.*

Definition B.10 (Weakly continuous)**.** Let X denote a Banach space. A functional $f \colon X \to \mathbb{R}$ is *weakly continuous* if

$$u_k \rightharpoonup u \text{ in } X \Rightarrow f(u_k) \to f(u)$$

for each weakly convergent sequence $\{u_k\} \subset X$.

Definition B.11 (Weakly lower semicontinuous)**.** Let X denote a Banach space. A functional $f \colon X \to \mathbb{R}$ is *weakly lower semicontinuous* if

$$u_k \rightharpoonup u \text{ in } X \Rightarrow \liminf_{k \to \infty} f(u_k) \geq f(u)$$

for each weakly convergent sequence $\{u_k\} \subset X$.

B.2.5 Embedding Theorems

For two Banach spaces X and Y a *continuous embedding* is denoted by $X \hookrightarrow Y$. We have the following results, e.g. from [Alt, 2012, Th. 8.6, Th. 8.9].

Lemma B.12. *Let $\Omega \subset \mathbb{R}^d$ be open and bounded, $0 \leq \lambda_1, \lambda_2 \leq 1$ and k_1, $k_2 \geq 0$. If Ω is Lipschitz for $k_1 > 0$ and $k_1 + \lambda_1 > k_2 + \lambda_2$, the embedding*

$$\mathcal{C}^{k_1,\lambda_1}(\bar{\Omega}) \hookrightarrow \mathcal{C}^{k_2,\lambda_2}(\bar{\Omega}) \text{ is compact.}$$

Lemma B.13. *Let $\Omega \subset \mathbb{R}^d$ be open, bounded and Lipschitz. Given integers k_1, $k_2 \geq 0$ and $1 \leq p_1, p_2 < \infty$, then*

(i) if $k_1 - \frac{d}{p_1} \geq k_2 - \frac{d}{p_2}$, $k_1 \geq k_2$

$$W^{k_1,p_1}(\Omega) \hookrightarrow W^{k_2,p_2}(\Omega)$$

(ii) if $k_1 - \frac{d}{p_1} > k_2 - \frac{d}{p_2}$, $k_1 > k_2$ the embedding is also compact.

This holds also if W^{k_1,p_1}, W^{k,p_2} are replaced by $W_0^{k_1,p_1}$ and $W_0^{k_2,p_2}$, respectively.

Embedding theorem 8.13 in [Alt, 2012] yields

Lemma B.14. *Let $\Omega \subset \mathbb{R}^d$ be open and bounded with Lipschitz boundary. We have that for integers $k \geq 1$, $m \geq 0$ and $1 \leq p < \infty$, $\lambda \in [0,1]$*

(i) if $k - \frac{d}{p} = m + \lambda$, $\lambda \neq 0, 1$

$$W^{k,p}(\Omega) \hookrightarrow \mathcal{C}^{m,\lambda}(\bar{\Omega})$$

(ii) if $k - \frac{d}{p} > m + \lambda$,

$$W^{k,p}(\Omega) \hookrightarrow \mathcal{C}^{m,\lambda}(\bar{\Omega}) \text{ compactly}$$

For any open and bounded set Ω the results hold for $W_0^{k,p}(\Omega)$ instead of $W^{k,p}(\Omega)$.

A function in $W^{1,p}(\Omega)$ can be identified with a continuous version in $\mathcal{C}(\bar{\Omega})$:

Lemma B.15 (Morrey's inequality)**.** *Assume $d < p \leq \infty$. Then there exists a constant C depending only on d and p such that for $\lambda := 1 - \frac{n}{p}$*

$$\|u\|_{\mathcal{C}^{0,\lambda}(\mathbb{R}^d)} \leq C\|u\|_{W^{1,p}(\mathbb{R}^d)} \quad \forall u \in \mathcal{C}^1(\mathbb{R}^d)\,. \tag{B.2}$$

In case of an open, bounded domain $\Omega \subset \mathbb{R}^d$ with a $\mathcal{C}^1$-boundary, an element $u \in W^{1,p}(\Omega)$ has a version $u^\star \in \mathcal{C}^{0,\lambda}(\bar{\Omega})$ with

$$\|u^\star\|_{\mathcal{C}^{0,\lambda}(\bar{\Omega})} \leq C\|u\|_{W^{1,p}(\mathbb{R}^d)} \text{ where } C \text{ depends only on } p, n, \Omega\,. \tag{B.3}$$

Proof. Theorem 4 and 5 in [Evans, 2010] in Chapter 5.6 on page 266 and following. □

B.3 Gâteaux- and Fréchet Differentiability

Definition B.16. Let X, Y be Banach spaces and U a nonempty open subset of X. Consider the map $F\colon U \subset X \to Y$.

(i) F is *directionally differentiable* at $u \in U$ if the limit

$$\mathrm{d}F(u;h) = \lim_{t \to 0^+} \frac{1}{t}(F(u+th) - F(u))$$

exists for all $h \in X$. In that case, $\mathrm{d}F(u;h)$ is called *directional derivative* of F at u in the direction of h. For a functional $J\colon U \to \mathbb{R}$ the directional derivative reduces to

$$\mathrm{d}J(u;h) = \mathrm{d}_t J(u+th)\Big|_{t=0}\,.$$

(ii) F is called *Gâteaux-differentiable* at $u \in U$ if F is directionally differentiable at u and there is $A \in \mathcal{L}(X, Y)$ such that

$$\mathrm{d}F(u; h) = Ah \text{ for all } h \in X\,.$$

The map $F' \colon u \mapsto F'(u) = A$ is the *Gâteaux-derivative*, and we say that $\mathrm{d}F(u; h) = F'(u)h$ is the *Gâteaux-differential* of F at u along h.

(iii) F is *Fréchet-differentiable* at $u \in U$ if there exists a linear operator $A \in \mathcal{L}(X, Y)$ such that

$$\|F(u + h) - F(u) - A(h)\|_Y = o(\|h\|_X)\,,$$

i.e. $\frac{\|F(u+h)-F(u)-A(h)\|_Y}{\|h\|_X} \to 0$ for $\|h\|_X \to 0$. If A exists it is uniquely determined and $A = F'(u)$. The map $F' \colon U \to \mathcal{L}(X, Y)$, $F' \colon u \mapsto F'(u)$ is called *Fréchet-derivative* of F. The quantity $\mathrm{d}F(u; h) = F'(u)h$ is called the *Fréchet-differential* of F at u along h.

(iv) Given a Fréchet-differentiable functional $J \colon U \to \mathbb{R}$ the gradient ∇J of J at u is the element of the dual space $X^* = \mathcal{L}(X, \mathbb{R})$ defined by

$$\langle \nabla J(u), h \rangle_{X^*, X} = \mathrm{d}J(u; h), \quad \forall h \in X\,.$$

If $X = H$ is an Hilbert space the Riesz representation theorem B.3 identifies $J'(u) \in H^*$ with an element of H

$$(\nabla J, h) = \mathrm{d}J(u; h), \quad \forall h \in H\,.$$

Remark B.17. Fréchet-differentiability Definition B.16, (iii) can also be stated as

$$F(u + h) = F(u) + F'(u)h + o(\|h\|_X)\,.$$

A useful characterization of Fréchet-differentiability is the following

Theorem B.18.
If $F \colon U \to Y$ is Gâteaux-differentiable in U and the Gâteaux-derivative

$$F'_G \colon U \to \mathcal{L}(X, Y),\ u \mapsto F'_G(u)$$

is continuous at u^*, then F is Fréchet-differentiable at u^* and $F'(u^*) = F'_G(u^*)$.

Proof. See [Ambrosetti and Prodi, 1993]. □

For both types of derivatives, Fréchet and Gâteaux, partial derivatives are defined as follows.

Definition B.19. Let X, Y, Z be Banach spaces and Q an open subset of $X \times Y$, $F \colon Q \to Z$. If for a fixed $(u^\star, v^\star) \in Q$, the map $F(u, v^\star) \colon X \to Z$ is Fréchet-differentiable at $u^\star$ w.r.t. u, the linear map $F'_u(u^\star, v^\star)$ is called *partial derivative* of F at $(u^\star, v^\star)$. Correspondingly, we denote the partial Fréchet-differential at $(u^\star, v^\star)$ along $h \in X$ by $\mathrm{d}_u F(u^\star, v^\star; h) = F'_u(u^\star, v^\star)h$.

Similarly define partial derivatives for v and fixed $u^\star$ and for Gâteaux-derivatives.

Lemma B.20 (Chain rule)**.** *For U and V open subsets of X and Y, respectively, let $F\colon U \to Y$ and $G\colon V \to Z$ with $F(U) \subset V$. If F is Fréchet-differentiable at $u \in U$ and G at $v := F(u) \in V$ then the composite map $H = G \circ F$ is Fréchet-differentiable at u and for $h \in U$*

$$H'(u)h = G'(v)\big(F'(u)h\big) \text{ or equally } \mathrm{d}H(u;h) = \mathrm{d}G\big(v;\mathrm{d}F(u;h)\big)\,.$$

Proof. See [Ambrosetti and Prodi, 1993, Prop. 1.4, p. 11]. □

Theorem B.21 (Implicit Function Theorem).
Let X, Y, Z be Banach spaces and let $F\colon G \to Z$ be a continuous Fréchet-differentiable map from an open set $G \subset X \times Y$ to Z. Let $(x^\star, y^\star) \in G$ be such that $F(x^\star, y^\star) = 0$ and that $F'_y(x^\star, y^\star) \in \mathcal{L}(Y, Z)$ has a bounded inverse. Then there exists an open neighborhood $U_X(x^\star) \times U_Y(y^\star) \subset G$ of $(x^\star, y^\star)$ and a unique continuous function $w\colon U_X(x^\star) \to Y$ such that

(i) $w(x^\star) = y^\star$

(ii) For all $x \in U_X(x^\star)$ there exists exactly one $y \in U_Y(y^\star)$ with $F(x, y) = 0$, namely $y = w(x)$.

(iii) The mapping w is continuously Fréchet-differentiable with derivative $w'(x) = F'_y(x, w(x))^{-1} F'_x((x, w(x))$.

Proof. This form of the implicit function theorem is stated in [Hinze et al., 2009, Theorem 1.41]. It can also be found in for instance in [Ambrosetti and Prodi, 1993, Theorem 2.3] together with a proof. □

C Geometry and Material Data

C.1 Material

The material properties in Table C.1 have been obtained from [MatWeb, 2015], where we converted them to the unit GPa and calculated the required parameters using

- $1\text{N}/\text{m}^2 = 1\text{Pa}$,
- $\lambda = K - \frac{2}{3}\mu$ with bulk modulus K and shear modulus μ or as in (2.22),
- $\mu = \frac{3}{2}(K - \lambda)$.

The parameters are given in units [GPa] common in engineering, differently to those in [Ciarlet, 1988, p. 129] where $[\text{kg}/\text{cm}^2]$ is used. For our computations we take the values according to the reference [Ciarlet, 1988], however, if one multiplies the values given there with the standard gravitation $g = 9.8\text{m}/\text{s}^2$, they are in the range of Table C.1.

material	E [GPa]	ν	λ [GPa]	μ [GPa]	database name
steel	200	0.29	106.7	80	AISI 1005 Low Carbon Steel
aluminum	68	0.36	64.3	25	Aluminum, Al

Table C.1: Parameters for different elastic materials from [MatWeb, 2015]

C.2 Surfaces

knot vector Ξ_0	$(0,0,1,1)$
knot vector Ξ_1	$(0,0,1,1)$
degree p_0	1
degree p_1	1
control points $(X_i)_i$	$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$
weights $(\omega_i)_i = 1$	
number of basis functions n_0	2
number of basis functions n_1	2

Table C.2: Right triangle

knot vector Ξ_0	$\begin{pmatrix}0,0,0,1,1,1\end{pmatrix}$
knot vector Ξ_1	$\begin{pmatrix}0,0,0,1,1,1\end{pmatrix}$
degree p_0	2
degree p_1	2
control points $(X_i)_i$	$\begin{pmatrix}-1 & -1 & 0 & -1 & 0 & 1 & 0 & 1 & 1\\ 0 & -1 & -1 & 1 & 0 & -1 & 1 & 1 & 0\end{pmatrix}$
weights $(\omega_i)_i$	$\begin{pmatrix}1 & \frac{\sqrt{2}}{2} & 1 & \frac{\sqrt{2}}{2} & 1 & \frac{\sqrt{2}}{2} & 1 & \frac{\sqrt{2}}{2} & 1\end{pmatrix}$
number of basis functions n_0	3
number of basis functions n_1	3

Table C.3: Circle at origin with radius 1

knot vector Ξ_0	0, 0, 0.5, 1, 1
knot vector Ξ_1	0, 0, 1, 1
degree p_0	1
degree p_1	1
control points $(X_i)_i$	$\begin{pmatrix}-1 & -0.5 & 0 & -4 & -4 & 0\\ 0 & 0.5 & 1 & 0 & 4 & 4\end{pmatrix}$
weights $(\omega_i)_i = 1$	
number of basis functions n_0	3
number of basis functions n_1	2

Table C.4: A quarter of the plate with hole

knot vector Ξ_0	$\begin{pmatrix}0,0,1,1\end{pmatrix}$
knot vector Ξ_1	$\begin{pmatrix}0,0,1,1\end{pmatrix}$
degree p_0	1
degree p_1	1
control points $(X_i)_i$	$\begin{pmatrix}0 & 1 & 0 & 1\\ 0 & 0 & 1 & 1\end{pmatrix}$
weights $(\omega_i)_i = 1$	
number of basis functions n_0	2
number of basis functions n_1	2

Table C.5: Square

Bibliography

[Adams and Fournier, 2003] Adams, R. and Fournier, J. (2003). *Sobolev Spaces*. Academic Press, 2nd edition.

[Allaire, 2007] Allaire, G. (2007). *Conception optimale de structures*, volume 58 of *Mathématiques et Applications*. Springer-Verlag Berlin Heidelberg, 1st edition.

[Allaire and Henrot, 2001] Allaire, G. and Henrot, A. (2001). On some recent advances in shape optimization. *Comptes Rendus de l'Académie des sciences Paris*, 329:383–396.

[Allaire and Jouve, 2005] Allaire, G. and Jouve, F. (2005). A level-set method for vibration and multiple loads structural optimization. *Computer Methods in Applied Mechanics and Engineering*, 194(30–33):3269–3290.

[Alt, 2012] Alt, H. W. (2012). *Lineare Funktionalanalysis*. Springer-Verlag Berlin Heidelberg, 6th edition.

[Ambrosetti and Prodi, 1993] Ambrosetti, A. and Prodi, G. (1993). *A primer of nonlinear analysis*. Number 34 in Cambridge Studies in advanced mathematics. Cambridge University Press.

[Atkinson and Han, 2001] Atkinson, K. E. and Han, W. (2001). *Theoretical Numerical Analysis. A Functional Analysis Framework* . Springer-Verlag New York.

[Bandara et al., 2015] Bandara, K., Cirak, F., Of, G., Steinbach, O., and Zapletal, J. (2015). Boundary element based multiresolution shape optimisation in electrostatics. *Journal of Computational Physics*, 297:584–598. accepted in Journal of Computational Physics.

[Bazilevs et al., 2006] Bazilevs, Y., Beirão da Veiga, L., Cottrell, J., Hughes, T. J. R., and Sangalli, G. (2006). Isogeometric Analysis: Approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16(7):1031–1090.

[Beirão da Veiga et al., 2014] Beirão da Veiga, L., Buffa, A., Sangalli, G., and Vázquez, R. (2014). Mathematical analysis of variational isogeometric methods. *Acta Numerica*, 23:157–287.

[Bendsøe and Sigmund, 2003] Bendsøe, M. P. and Sigmund, O. (2003). *Topology optimization: Theory, Methods and Applications*. Springer-Verlag Berlin Heidelberg.

[Berggren, 2010] Berggren, M. (2010). A unified discrete–continuous sensitivity analysis method for shape optimization. In Fitzgibbon, W., Kuznetsov, Y., Neittaanmäki, P., Périaux, J., and Pironneau, O., editors, *Applied and Numerical Partial Differential Equations*, volume 15 of *Computational Methods in Applied Sciences*, chapter 4, pages 25–39. Springer Netherlands.

[Blanchard et al., 2013] Blanchard, L., Duvigneau, R., Vuong, A.-V., and Simeon, B. (2013). Shape Gradient for Isogeometric Structural Design. *Journal of Optimization Theory and Applications*, 161(2):1–7.

[Boeing, 2015] Boeing (2015). Boeing 787 dreamliner. Look up date 2015-05-22: `www.boeing.com/commercial/787`.

[Brandenburg, 2011] Brandenburg, C. (2011). *Adjoint-Based Adaptive Multilevel Shape Optimization based on Goal-Oriented Error Estimators for the Instationary Navier-Stokes Equations*. PhD thesis, Fachbereich Mathematik, Technische Universität Darmstadt, Germany.

[Brandenburg et al., 2009] Brandenburg, C., Lindemann, F., Ulbrich, M., and Ulbrich, S. (2009). A continuous adjoint approach to shape optimization for navier stokes flow. In Kunisch, K., Sprekels, J., Leugering, G., and Tröltzsch, F., editors, *Optimal Control of Coupled Systems of Partial Differential Equations*, volume 158 of *International Series of Numerical Mathematics*, pages 35–56. Birkhäuser Basel.

[Brenner and Scott, 2008] Brenner, S. C. and Scott, L. R. (2008). *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer-Verlag New York, 3rd edition.

[Budd et al., 2009] Budd, C. J., Huang, W., and Russel, R. D. (2009). Adaptivity with moving grids. *Acta Numerica*, pages 1–131.

[Carnarius et al., 2010] Carnarius, A., Thiele, F., Özkaya, E., and Gauger, N. R. (2010). Adjoint approaches for optimal flow control. *American Institute of Aeronautics and Astronautics Journal*, 5088.

[Ciarlet, 1979] Ciarlet, P. (1979). *The Finite Element Method For Elliptic Problems*. Elsevier.

[Ciarlet, 1988] Ciarlet, P. (1988). *Mathematical Elasticity. Volume I: Three-dimensional Elasticity*, volume 20 of *Studies in Mathematics and Its Applications*. Elsevier.

[Clune et al., 2014] Clune, R., Kelliher, D., Robinson, J. C., and Campbell, J. S. (2014). NURBS modeling and structural shape optimization of cardiovascular stents. *Structural and Multidisciplinary Optimization*, 50(1):159–168.

[Cohen et al., 2010] Cohen, E., Martin, T., Kirby, R., Lyche, T., and Riesenfeld, R. (2010). Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199:334–356.

[De Boor, 2001] De Boor, C. (2001). *A practical guide to splines*, volume 27. Springer-Verlag New York.

[Delfour and Zolésio, 2011] Delfour, M. C. and Zolésio, J.-P. (2011). *Shapes and Geometries. Metrics, Analysis, Differential Calculus, and Optimization*. Advances in Design and Control. SIAM, Philadelphia, 2nd edition.

[Dido, 2001] Dido (2001). thank you (cd): thank you. `http://www.songtexte.com/songtext/dido/thankyou-4bd6aba2.html`. [Online; accessed 23-June-2015].

[Dubois, 2005] Dubois, P. F. (2005). Maintaining correctness in scientific programs. *Computing in Science & Engineering*.

[Eppler et al., 2007] Eppler, K., Harbrecht, H., and Schneider, R. (2007). On convergence in elliptic shape optimization. *SIAM Journal on Control and Optimization*, 46(1):61–83.

[Evans, 2010] Evans, L. C. (2010). *Partial Differential Equations*. AMS, 2nd edition.

[Forsey and Bartels, 1988] Forsey, D. R. and Bartels, R. H. (1988). Hierarchical B-Spline Refinement. *Computer Graphics*, 22(4):205–212.

[Fußeder and Simeon, 2015] Fußeder, D. and Simeon, B. (2015). Algorithmic aspects of isogeometric shape optimization. accepted in Isogeometric Analysis and Applications - IGAA 2014, Lecture Notes in Computational Science and Engineering.

[Fußeder et al., 2015] Fußeder, D., Vuong, A.-V., and Simeon, B. (2015). Fundamental aspects of shape optimization in the context of isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 286(39):313–331.

[Gautschi, 1993] Gautschi, W. (1993). Gauss-type quadrature rules for rational functions. http://arxiv.org/abs/math/9307223.

[Gekeler, 2010] Gekeler, E. (2010). *Mathematische Methoden zur Mechanik*. Springer-Verlag Berlin Heidelberg, 2nd edition.

[Giannelli et al., 2012] Giannelli, C., Jüttler, B., and Speleers, H. (2012). THB-splines: The truncated basis for hierarchical splines. *Computer Aided Geometric Design*, 29:485–498.

[Ginnis et al., 2013] Ginnis, A., Duvigneau, R., Politis, C., Kostas, K., Gerostathis, T., and D, K. P. (2013). A multi-objective optimization environment for ship-hull design based on a bem-isogeometric solver. In *The fifth Conference on Computational Methods in Marine Engineering (Marine 2013), Hamburg, Germany*.

[Haslinger and Mäkinen, 2003] Haslinger, J. and Mäkinen, R. A. E. (2003). *Introduction to Shape Optimization: Theory, Approximation, and Computation*. Advances in Desgin and Control. SIAM.

[Hicken and Zingg, 2008] Hicken, J. E. and Zingg, D. W. (2008). Integrated geometry parametrization and grid movement using b-spline meshes. In *Proceedings of The 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. AIAA-2008-6079, Victoria, British Columbia, Canada.

[Hinze et al., 2009] Hinze, M., Pinnau, R., Ulbrich, M., and Ulbrich, S. (2009). *Optimization with PDE Constraints*, volume 23 of *Mathematical Modelling: Theory and Applications*. Springer Netherlands.

[Hughes et al., 2005] Hughes, T. J. R., Cottrell, J., and Bazilevs, Y. (2005). Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39):4135–4195.

[Hughes et al., 2006] Hughes, T. J. R., Cottrell, J., Reali, A., and Bazilevs, Y. (2006). Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195:5257–5296.

[Hughes et al., 2010] Hughes, T. J. R., Reali, A., and Sangalli, G. (2010). Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199:301–313.

[Ito and Kunisch, 2008] Ito, K. and Kunisch, K. (2008). *Lagrange Multiplier Approach to Variational Problems and Applications*. SIAM.

[Ito et al., 2008] Ito, K., Kunisch, K., and Peichl, G. (2008). Variational approach to shape derivatives. *ESAIM: Control, Optimisation and Calculus of Variations*, 14(3):517.

[Johnson and Tezduyar, 1994] Johnson, A. A. and Tezduyar, T. E. (1994). Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering*, 119:73–94.

[Johnson, 2014] Johnson, S. G. (2014). The NLopt nonlinear-optimization package. `http://ab-initio.mit.edu/nlopt`. [Online; accessed 10-October-2014].

[Jüttler et al., 2014] Jüttler, B., Kapl, M., Nguyen, D.-M., Pan, Q., and Pauley, M. (2014). Isogeometric segmentation: The case of contractible solids without non-convex edges. *Computer-Aided Design*, 57(0):74–90.

[Kiendl et al., 2014] Kiendl, J., Schmidt, R., Wüchner, R., and Bletzinger, K.-U. (2014). Isogeometric shape optimization of shells using semi-analytical sensitivity analysis and sensitivity weighting. *Computer Methods in Applied Mechanics and Engineering*, 274:148–167.

[Kiniger, 2015] Kiniger, B. (2015). *Error estimates for finite element methods in shape optimization*. PhD thesis, Technische Universität München, Germany.

[Kiniger and Vexler, 2013] Kiniger, B. and Vexler, B. (2013). A priori error estimates for finite element discretizations of a shape optimization problem. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47(6):1733 – 1763.

[Lindemann, 2012] Lindemann, F. (2012). *Theoretical and Numerical Aspects of Shape Optimization with Navier-Stokes Flows*. PhD thesis, Technische Universität München, Germany.

[MATLAB, 2012] MATLAB (2012). Release 2012a. The MathWorks Inc.

[MatWeb, 2015] MatWeb (1996-2015). Material property data. [Online; accessed 10-June-2015].

[Murat and Simon, 1976a] Murat, F. and Simon, J. (1976a). Etude de problèmes d'optimal design. In Céa, J., editor, *Optimization Techniques Modeling and Optimization in the Service of Man Part 2*, number 41 in Lecture Notes in Computer Sciences, pages 54–62. Springer, Berlin/Heidelberg.

[Murat and Simon, 1976b] Murat, F. and Simon, J. (1976b). Sur le contrôle par un domaine géométrique. Technical Report no 76015, Laboratoire d'Analyse Numérique, Université de Paris 6.

[Nečas, 2012] Nečas, J. (2012). *Direct Methods in the Theory of Elliptic Equations*. Springer-Verlag Berlin Heidelberg.

[Neittaanmäki and Salmenjoki, 1989] Neittaanmäki, P. and Salmenjoki, K. (1989). Sensitivity analysis for optimal shape design problems. *Structural Optimization*, 1:241–251.

[Nguyen, 2012] Nguyen, D.-M. (2012). *Isogeometric Analysis and Shape Optimization in Electromagnetism*. PhD thesis, Technical University of Denmark.

[Nguyen et al., 2011] Nguyen, D.-M., Evgrafov, A., Gersborg, A. R., and Gravesen, J. (2011). Isogeometric shape optimization of vibrating membranes. *Computer Methods in Applied Mechanics and Engineering*, 200(13):1343–1353.

[Nguyen et al., 2012] Nguyen, D.-M., Evgrafov, A., and Gravesen, J. (2012). Isogeometric shape optimization for electromagnetic scattering problems. *Progress in Electromagnetics Research B*, 45:117–146.

[Nocedal and Write, 2006] Nocedal, J. and Write, S. J. (2006). *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer-Verlag New York, 2nd edition.

[Nochetto et al., 2009] Nochetto, R. H., Siebert, K. G., and Veeser, A. (2009). Theory of adaptive finite element methods: An introduction. In DeVore, R. and Kunoth, A., editors, *Multiscale, Nonlinear and Adaptive Approximation*, pages 409–542. Springer Berlin Heidelberg.

[Nørtoft and Gravesen, 2013] Nørtoft, P. and Gravesen, J. (2013). Isogeometric shape optimization in fluid mechanics. *Structural and Multidisciplinary Optimization*, 48:909–925. DOI 10.1007/s00158-013-0931-8.

[Piegl and Tiller, 1995] Piegl, L. and Tiller, W. (1995). *The NURBS book*. Monographs in Visual Communication. Springer-Verlag Berlin Heidelberg.

[Pironneau, 1983] Pironneau, O. (1983). *Optimal Shape Design for Elliptic Systems*. Springer Series in Computational Physics. Springer-Verlag New York.

[Protas et al., 2004] Protas, B., Bewley, T. R., and Hagen, G. (2004). A computational framework for the regularization of adjoint analysis in multiscale PDE systems. *Journal of Computational Physics*, 195:49–89.

[Prudhomme and Oden, 1999] Prudhomme, S. and Oden, J. T. (1999). On goal-oriented error estimation for elliptic problems: application to the control of pointwise errors. *Computer Methods in Applied Mechanics and Engineering*, 176(1):313–331.

[Qian, 2010] Qian, X. (2010). Full analytical sensitivities in NURBS based isogeometric shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 199:2059–2071.

[Quarteroni and Valli, 2008] Quarteroni, A. and Valli, A. (2008). *Numerical approximation of partial differential equations*, volume 23 of *Springer Series in Computational Mathematics*. Springer Berlin Heidelberg.

[Sala et al., 2004] Sala, M., Heroux, M. A., Day, D. M., and Willenbring, J. M. (2004). Trilinos tutorial. Technical Report SAND2004-2189, Sandia National Laboratories.

[Schmidt et al., 2011] Schmidt, S., Illic, C., Volker, S., and Gauger, N. R. (2011). Airfoil design for compressible inviscid flow based on shape calculus. *Optimization and Engineering*, 12(3):349–369.

[Schulz, 2014] Schulz, V. (2014). A Riemannian View on Shape Optimization. *Foundations of Computational Mathematics*, 14(3):483–501.

[Schulz and Borzì, 2012] Schulz, V. and Borzì, A. (2012). *Computational Optimization of Systems Governed by Partial Differential Equations*. Computational Science & Engineering. SIAM.

[Schumaker, 1981] Schumaker, L. L. (1981). *Spline Functions: Basic Theory*. Wiley, New York.

[Sokolowski and Zolésio, 1992] Sokolowski, J. and Zolésio, J.-P. (1992). *Introduction to Shape Optimization. Shape Sensitivity Analysis*, volume 16 of *Springer Series in Computational Mathematics*. Springer-Verlag Berlin Heidelberg.

[Svanberg, 2002] Svanberg, K. (2002). A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM Journal on Optimization*, 12(2):555–573.

[Takacs and Takacs, 2015] Takacs, S. and Takacs, T. (2015). Approximation error estimates and inverse inequalities for b-splines of maximum smoothness. Presented at IGA 2015 in Trondheim, Norway. [arXiv:1502.03733].

[Takacs and Jüttler, 2012] Takacs, T. and Jüttler, B. (2012). H^2 regularity properties of singular parameterizations in isogeometric analysis. *Graphical Models*, 74:361–372.

[Timoshenko and Goodier, 1951] Timoshenko, S. and Goodier, J. (1951). *Theory of elasticity*. McGraw-Hill, New York.

[Tröltzsch, 2010] Tröltzsch, F. (2010). *Optimal control of partial differential equations. Theory, methods and applications*, volume 112 of *Graduate Studies in Mathematics*. AMS.

[Ulbrich and Ulbrich, 2012] Ulbrich, M. and Ulbrich, S. (2012). *Nichtlineare Optimierung*. Mathematik kompakt. Birkhäuser Basel, 1st edition.

[van Keulen et al., 2005] van Keulen, F., Hafka, R. T., and Kim, N. H. (2005). Review of options for structural design sensitivity analysis. Part 1: Linear systems. *Computer Methods in Applied Mechanics and Engineering*, 194:3213–3243.

[Varberg, 1971] Varberg, D. E. (1971). Change of variables in multiple integrals. *The American Mathematical Monthly*, 78(1):42–45.

[Vuong, 2012] Vuong, A.-V. (2012). *Adaptive Hierarchical Isogeometric Finite Element Methods*. Vieweg+Teubner Verlag.

[Vuong and Fußeder, 2013] Vuong, A.-V. and Fußeder, D. (2013). On local hierarchical refinement and error estimation for isogeometric analysis. *Proceedings of Applied Mathematics and Mechanics*, 13(1):565–568.

[Vuong et al., 2011] Vuong, A.-V., Giannelli, C., Jüttler, B., and Simeon, B. (2011). A hierarchical approach to adaptive local refinement in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*.

[Vuong et al., 2010] Vuong, A.-V., Heinrich, C., and Simeon, B. (2010). Isogat: A 2d tutorial matlab code for isogeometric analysis. *Computer Aided Geometric Design*, 27:644–655. Advances in Applied Geometry.

[Wächter and Biegler, 2006] Wächter, A. and Biegler, L. T. (2006). On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical Programming*, 106(1):25–57.

[Wall et al., 2008] Wall, W., Frenzel, M., and Cyron, C. (2008). Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197:2976–2988.

[Xu et al., 2013] Xu, G., Mourrain, B., Duvigneau, R., and Galligo, A. (2013). Optimal analysis-aware parameterization of computational domain in 3D isogeometric analysis. *Computer-Aided Design*, 45(4):812–821.

Akademischer Lebenslauf

seit Dez. 2011	Technische Universität Kaiserslautern *wissenschaftliche Mitarbeiterin*
Okt. 2006 – Sept. 2011	Technische Universität München *Diplom Mathematik*
Sept. 2004 – Aug. 2006	Staatliche Berufsoberschule Technik München *Allgemeine Hochschulreife*

Curriculum Vitae

since Dec. 2011	University of Kaiserslautern, Germany *Scientific assistant*
Oct. 2006 – Sept. 2011	Technische Universität München, Germany *Diploma in mathematics*
Sept. 2004 – Aug. 2006	Staatliche Berufsoberschule Technik München, Germany *University entrance qualification*